Running ahead of evolution - AI based simulation for predicting future high-risk SARS-CoV-2 variants

Jie Chen[1,2,†], Zhiwei Nie[2,1,†], Yu Wang[1,†], Kai Wang[1,†], Fan Xu[1,†], Zhiheng Hu[1], Bin Zheng[1], Zhennan Wang[1], Guoli Song[1], Jingyi Zhang[1], Jie Fu[3], Xiansong Huang[1], Zhongqi Wang[1], Zhixiang Ren[1], Qiankun Wang[1], Daixi Li[1], Dongqing Wei[1], Bin Zhou[4,1,#], Chao Yang[5,#], Yonghong Tian[6,2,1,#]
[1]Peng Cheng Laboratory, China
[2]School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[3]Beijing Academy of Artificial Intelligence
[4]School of Information Science and Engineering, Shandong University, Qingdao, China
Email: binzhou@sdu.edu.cn
[5]School of Mathematical Sciences, Peking University, China
Email: chao_yang@pku.edu.cn
[6]School of Computer Science, Peking University, China
Email: yhtian@pku.edu.cn

Abstract

The never-ending emergence of SARS-CoV-2 variations of concern (VOCs) has challenged the whole world for pandemic control. In order to develop effective drugs and vaccines, one needs to efficiently simulate SARS-CoV-2 spike receptor binding domain (RBD) mutations and identify high-risk variants. We pretrain a large protein language model on approximately 408 million protein sequences and construct a high-throughput screening for the prediction of binding affinity and antibody escape. As the first work on SARS-CoV-2 RBD mutation simulation, we successfully identify mutations in the RBD regions of 5 VOCs and can screen millions of potential variants in seconds. Our workflow scales to 4096 NPUs with 96.5% scalability and 493.9× speedup in mixed precision computing, while achieving a peak performance of 366.8 PFLOPS (reaching 34.9% theoretical peak) on Pengcheng Cloudbrain-II. Our method paves the way for simulating coronavirus evolution in order to prepare for a future pandemic that will inevitably take place.

Keywords: ***COVID-19, virus mutation simulation, high-risk SARS-CoV-2 variant prediction, deep learning, language model, pretraining, iterative fine-tuning***

[†]Authors with symbol indicate equal contribution.
[#]Corresponding authors.

1 JUSTIFICATION FOR PRIZE

We develop a novel multi-constraint variation prediction framework to simulate SARS-CoV-2 RBD mutations, reaching a peak performance of 366.8 PFLOPS with 96.5% scalability and achieving 493.9× speedup. Our method facilitates the prediction and prioritization of future high-risk variants for the early deployment of drugs and vaccines.

2 PERFORMANCE ATTRIBUTES

Performance attribute	Our submission
Category of achievement	time-to-solution, scalability
Type of method used	machine learning
Results reported for	whole application using and except I/O
Precision reported	mixed precision
System scale	results measured on full-scale system
Measurement mechanism	timers, FLOP count, performance modeling

3 OVERVIEW OF THE PROBLEM

Coronavirus Disease 2019 (COVID-19) has spread rapidly to more than 200 countries or regions since December 2019. Due to its high infectivity, there have been over 617 million confirmed cases, including approximately 6.5 million deaths, reported by the World Health Organization (WHO) as of 8 October 2022 (https://covid19.who.int/). In addition to being a serious threat to human health, COVID-19 has had a catastrophic impact on the global economy.

The virus that causes the pandemic is the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) (Figure 1a), which belongs to the genus Betacoronavirus and has nearly 80% sequence similarity with the severe acute respiratory syndrome coronavirus (SARS-CoV) [1–3].

As the pandemic enters its third year, SARS-CoV-2 has been creating waves of infections across the globe (Figure 1b,c) [4] due to the high mutation rate of this RNA virus. Which potential SARS-CoV-2 variants may become the next VOCs? Do we need to develop new vaccines to deal with new variants? In which direction will the virus evolve? Shall we just give up as a society and hope that the virus will finally fade away? These are the **inconvenient questions** that every country on this planet must answer.

Before the current pandemic, the best-known Betacoronaviruses are SARS-CoV and Middle East respiratory syndrome coronavirus (MERS-CoV), which have relatively more severe clinical symptoms than most coronaviruses, which can infect humans but cause only mild symptoms [5–9]. In the past two decades, the viruses mentioned above have led to two epidemics: SARS (2002) and MERS (2012) [9]. SARS-CoV-2 can also infect the human respiratory system, but has a much higher infection rate than that of SARS-CoV or MERS-CoV [10, 11].

Three sets of proteins, including structural proteins, nonstructural proteins, and accessory proteins, are encoded by SARS-CoV-2 [1] (Figure 1a). There are four main classes of structural proteins, namely, spike protein (S), nucleocapsid protein (N), membrane protein (M) and envelope protein (E), which support the structure of the virus in terms of shape or function [1, 12]. In particular, in addition to their high similarity in sequences, SARS-CoV-2 and SARS-CoV have the same mechanism of infecting host cells, that is, binding to the host entry receptor angiotensin-converting enzyme 2 (hACE2) [3,13–15]. During infection, the trimeric S protein is cleaved by host proteases into the N-terminal S1 subunit and the C-terminal S2 subunit. The receptor-binding domain (RBD) is an important component of the S1 subunit (Figure 1a) that is responsible for binding to ACE2, and is the primary binding target for neutralizing antibodies (NAbs) [11, 16–18]. Therefore, the S protein plays a key role in viral infection and the immunne evasion process [19, 20].

SARS-CoV-2 continues to mutate with a high mutation rate [21] and has evolved into five main variants of concern (VOCs) as of May 2022: B.1.1.7 (Alpha), B.1.351 (Beta), P.1 (Gamma), B.1.617.2 (Delta) and B.1.1.529 (Omicron) (https://www.who.int/en/activities/tracking-SARS-CoV-2-variants/, Figure 1b,c). These SARS-CoV-2 variants with novel spike protein mutations have created waves of infections and reinfections across the globe (Figure 1d). It's vitally important to identify early [22] or, even better, to predict dangerous viral mutations that may enhance viral fitness including binding affinity, viral infectivity, or immunity escape.

The Global Initiative on Sharing All Influenza Data (GISAID) (https://www.gisaid.org/) [23] has recorded more than 13 million SARS-CoV-2 genomes submitted by scientists around the world. This large number of genomic sequences presents an excellent opportunity to study the spread and evolution of SARS-CoV-2. Computational methods such as the Gillespie algorithms can be used to simulate realistic substitution patterns of large-scale closely related genomic datasets, e.g., simulators targeting gene trees, ancestral recombination graphs, or phylogenetic trees [24–32]. Deep learning (DL) models can also learn hidden evolution patterns from the massive number of submitted virus sequences, thus prioritizing potential future viral mutations that might introduce the next VOCs [33, 34].

As shown in Figure 1a, the RBD region of the spike protein is an area of concern because it has a high mutation rate, which can significantly affect binding to hACE2, as well as antibodies. In this work, we simulate RBD mutations by learning, generating, screening, and fine-tuning DL models based on pretrained protein language models as shown in Figure 1e. A multi-constrains variation prediction framework(MCVP) is designed to learn from millions of RBD sequences and experimental measurements of binding affinity between single RBD mutations and hACE2/antibodies. MCVP utilizes active learning based on a pretrained protein language model. This HPC-driven work can evaluate RBD mutations based on protein expression, binding affinity, antibody escape to ultimately provide assistance in the battle against SARS-CoV-2.

4 CURRENT STATE OF THE ART

4.1 Predictive modeling of SARS-CoV-2 variants

During the pandemic, studies with a variety of focuses and models for predicting the mutation of SARS-CoV-2 have emerged. For example, a renewal-equation-based model was used to describe the adaptive evolution among multiple variants of SARS-CoV-2 including R.1, Alpha, and Delta, and then to predict the dominant variants in Japan before the start of the Tokyo Olympic Games [35]. Furthermore, some works seeked to accurately predict the fitness of SARS-CoV-2 variants, which was used to characterize how efficiently the virus

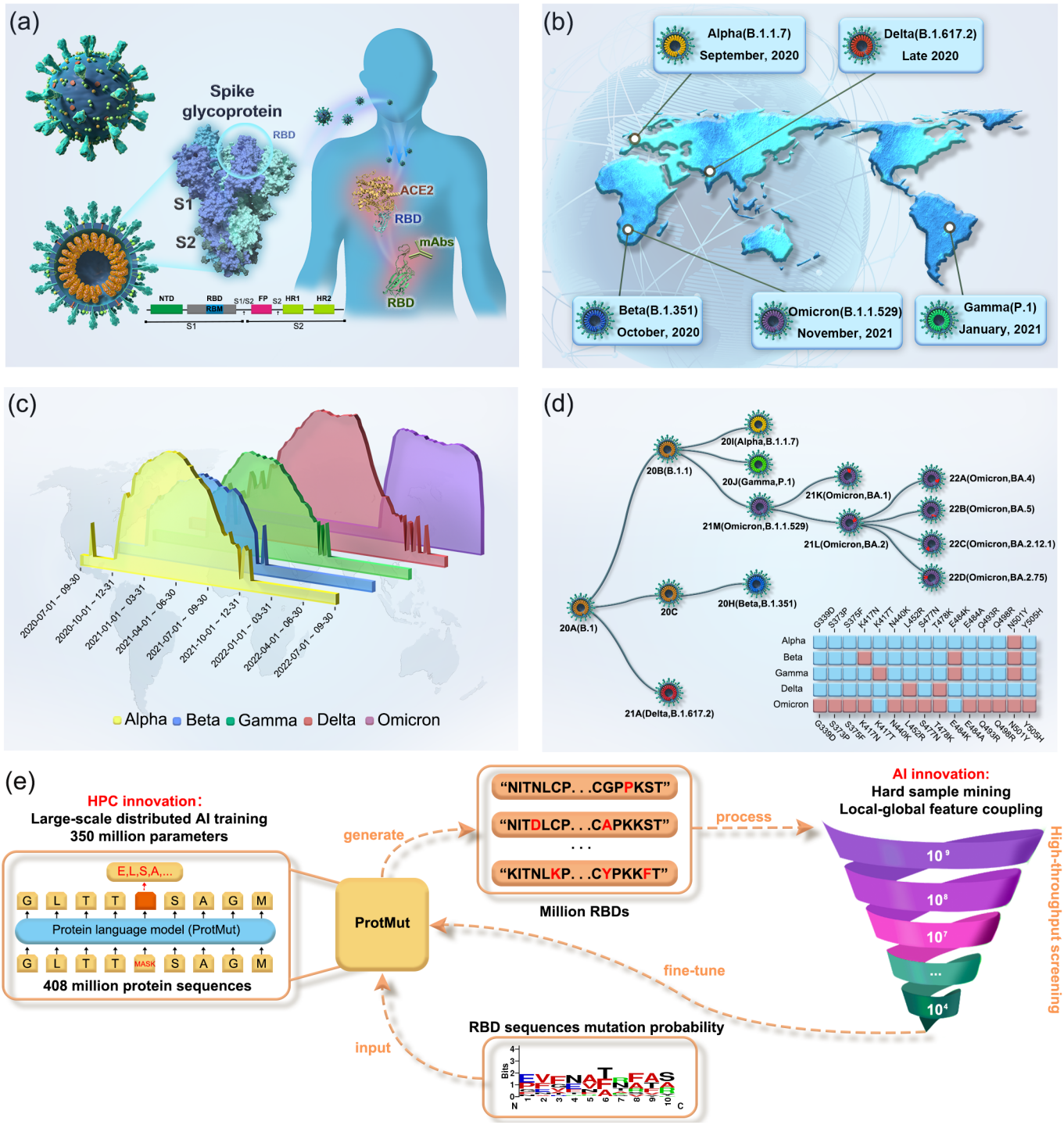

Figure 1: Overview of the problem and our solution. a) The structural diagram of SARS-CoV-2, in which the RBD on the spike protein is an important region to which hACE2 and the majority of neutralizing antibodies bind to. b) The approximate detection time and places of the five VOCs (Alpha, Beta, Gamma, Delta, and Omicron). c) Waves of infections caused by the five VOCs from the outbreak of COVID-19 to the present. d) The phylogenetic tree of SARS-CoV-2 VOCs and the comparison of the variation sites of the five VOCs in the RBD regions. e) Our methodology for simulating the viral mutation in the RBD. With the support of an HPC optimization strategy that integrates software and hardware, a protein language model (ProtMut) is efficiently pretrained for the generation of RBD mutations. With reference to the mutation frequency of each mutation site in the RBD in the real world, ProtMut can generate billions of RBD variants. These variants are sequentially screened for binding affinity with hACE2, and antibody escape capability. The screened variants are used to fine-tune the ProtMut generator. The retrained ProtMut model is more likely to generate viral variants with better binding to hACE2, better capability for antibody escape, and stronger fitness.

produces infectious progeny. A computational model named SpikePro [36] was designed to predict the fitness of SARS-CoV-2 from the sequence and structure of the spike protein in order to allow the identification of new dangerous variants. PyR_0 [22], a hierarchical Bayesian multinomial logistic regression model, was developed to infer lineage fitness, forecast future lineage proportions, and identify mutations relevant to fitness.

DL methods have recently been shown to perform well in predicting coronavirus mutations. Specifically, a three-dimensional convolutional neural network (3D CNN) based on spike dinucleotide composition representation was used to learn the human adaptation of existing coronaviruses and predict the adaptation of SARS-CoV-2 VOCs [37]. DL language models have also been applied for protein prediction tasks, as common protein motifs and domains can be analogized to words, phrases, and sentences in human language [38–41]. A Transformer-based discriminative model was trained with SARS-CoV-2 genetic sequences to predict potential mutations that may lead to enhanced virus transmissibility [42]. Motivated by the success of masked language models such as BERT [43], we design a pretrained protein language model for comprehensive variant prediction, aiming to simulate circulating viral mutation and predict of potentially risky variants. In this work, we pretrain our protein language model on a large-scale set of protein sequences using a supercomputer with exascale AI/DL training capabilities and further perform fine-tuning and multiconstraint screening on RBD sequences of the spike protein in SARS-CoV-2 to generate possible future variant branches.

4.2 Large-scale language model training

The existing state-of-the-art DL language models, especially various BERT variations [43–47] with Transformer as the core, have achieved outstanding performance in many fields. Recently, some works have emerged with a focus on transferring language models to large-scale protein representation learning, e.g., ESM [48] and ProtTrans [49], which were trained on the Summit supercomputer, and demonstrated that large-scale pretrained language models can capture latent grammar of protein sequences to a certain degree [49].

Minibatch stochastic gradient descent has been found to be very effective for large-scale learning [50]. However, updating the parameters in small batches makes optimization unstable [51]. For large-scale datasets, large-batch training with data parallelism has found increasing popularity [46], as it can improve the data communication and hardware utilization of a model. However, how to set the best batch size is a complex optimization problem. Some works [52–55] have reported that increasing the batch size beyond a certain point can result in poor generalization performance.

5 INNOVATIONS REALIZED

5.1 Overview of MCVP

MCVP is a heterogeneous system for simulating the effect of the RBD mutations on the fitness of SARS-COV-2 viruses. This system includes 1) a pretrained protein language generative model for RBD mutation generation, 2) an RBD and hACE2 binding affinity prediction model for selecting RBD mutants that have higher binding affinities than the wild type, and 3) an immune escape prediction model for selecting RBD mutants that are more likely to evade antibody attacks.

The training and validation data for the system are collected from various authoritative resources. We download protein sequences from the UniRef database [56] for the training of the protein language model. We download SARS-COV-2 genomes from the GISAID database[1], which includes more than twelve million genome sequences of SARS-CoV-2 for rapidly sharing. The S protein sequences are obtained from this platform, then the RBD region sequences are segmented for model fine-tuning and analyzed for the probability of the mutation rate at each position. SARS-COV-2 VOC defining mutations are downloaded from https://cov-lineages.org/index.html#global_reports.

5.1.1 The workflow of MCVP

We design MCVP to follow the workflow as shown in Figure 2a. The first module of the MCVP is a BERT-based language model, hereafter called ProtMut. ProtMut is trained with the UniRef90 dataset, including approximately 144 million protein sequences. All protein sequences are chopped into lengths of 256, as the RBD region of the spike protein S1 consists of 201 amino acids within the location range of 331-531 [57]. BERT is a bidirectional model for natural language processing that attempts to reconstruct corrupted tokens. For protein language modeling, 15% of each input protein sequence is masked. During the training process, ProtMut reconstruct the masked amino acids. The details of ProtMut can refer to Appendix A. At the end of the training, ProtMut has learned protein embeddings that captured some of the biophysical features of the protein sequences.

We use ProtMut in two ways. First, we design an RBD-variation-generating module based on ProtMut. We fine-tune ProtMut with RBD sequences truncated

[1] https://gisaid.org/

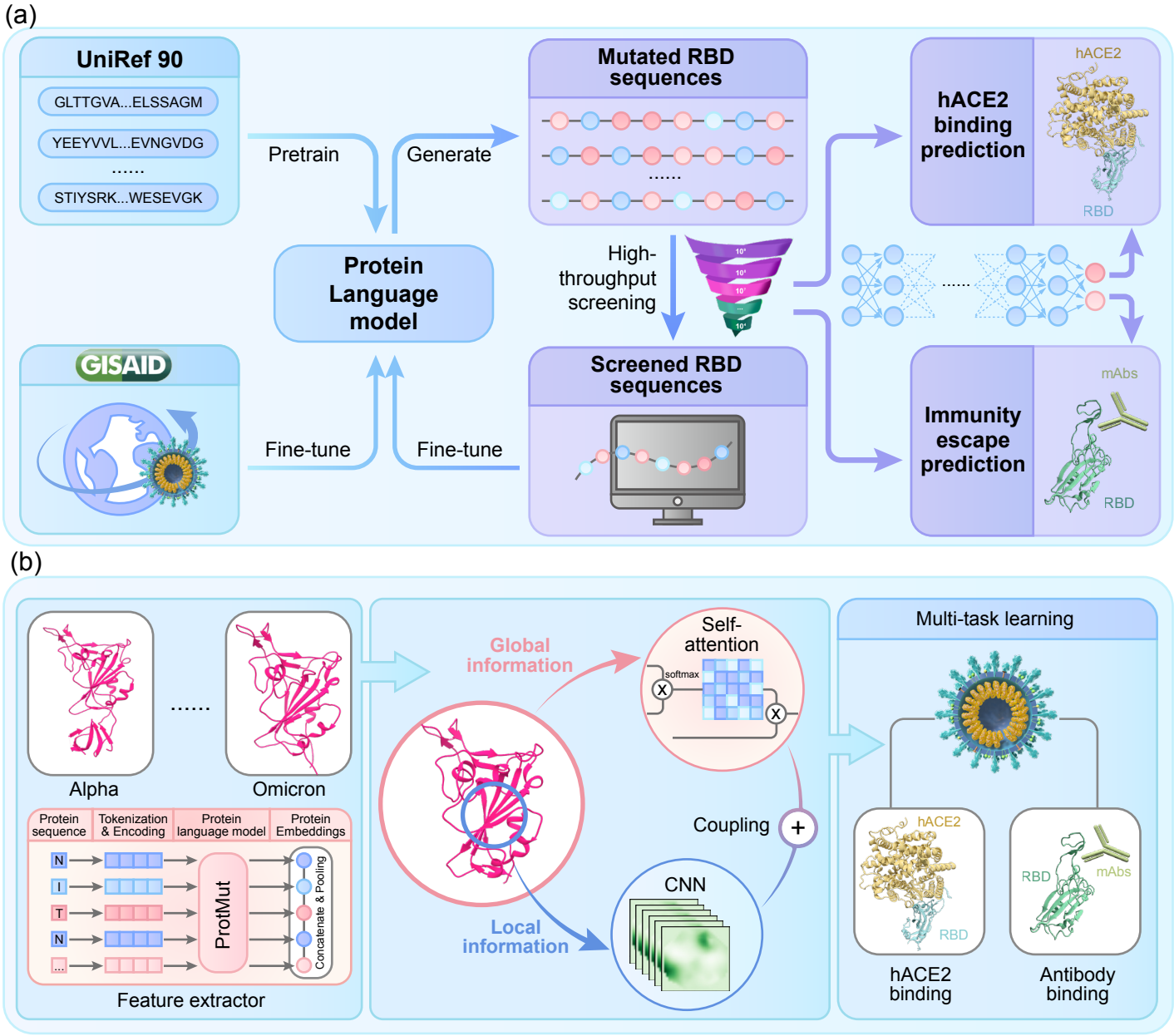

Figure 2: a) The workflow of MCVP. It consists of four modules, including pretraining, fine-tuning, generation, and high-throughput screening. b) Two transfer-learning models for high-throughput screening. Three modules make up the whole processing workflow: a feature extractor module, a feature refinement module, and a downstream task module. The protein embeddings learned by ProtMut are further refined through the coupling of global and local features. And then neural networks are trained for two different downstream tasks.

from the spike protein sequences which were downloaded from GISAID. Afterward, we generate new RBD mutations by generating missing amino acids from a masked RBD sequence selected as the starting sequence. Second, as a protein embedding generator, ProtMut provide meaningful vector representations of RBD mutations. These embeddings are used as the inputs to a binding affinity prediction model(BindTrans), and an immunity escape prediction model (EscTrans). The above models, BindTrans and EscTrans (manuscripts in preparation), are essential in selecting RBD mutations that are more advantageous in the sense of virus fitness and survival because of higher binding affinities and immune evasion.

We use ProtMut to generate millions of RBD mutations with Pengcheng Cloudbrain-II. Subsequently, two DL filters are used to screen the various generated variants of the RBD based on region binding affinity and immunity escape in a high-throughput manner. The *in*

silico screening is designed to simulate the evolution of SARS-CoV-2 in nature. Therefore, the variants that passed this screening could be considered as evolutionarily more advantageous. After we have completed one round of mutation simulation, the selected variants are used as trainging examples for ProtMut to fine-tune the mutation model, which forces the model to learn the characteristics of those variations that are more likely to survive the evolutionary selection. By repeating this procedure, ProtMut is guided to generate variants that were more likely to have evolutionary advantages, thus enabling the simulation of SARS-CoV-2 RBD mutation generation.

As shown in Figure 2b, the protein embedding generation process started with tokenization of a protein sequence and addition of the positional encoding. The resulting vectors are passed through ProtMut to create context-aware embeddings for each amino acid, which are the last hidden state of the Transformer's attention stack. Then these embeddings are concatenated and pooled along the length-dimension to obtain a fixed-size embedding irrespective of the length of the input protein sequence. In MCVP, two DL predictors are developed, based on the sequence embeddings extracted by ProtMut. The first is a binding affinity predictor designed for forecasting changes in binding affinity between the mutated RBD and hACE2. The second predictor can be used to evaluate the comprehensive antibody escape capability of the variants through antibody escape prediction.

5.1.2 Generation of variants

A variant generation module is designed based on the ProtMut model. Essentially, we assume that the ProtMut model has learned the general properties of proteins through self-supervised learning on billions of protein sequences. Then, by fine-tuning ProtMut on millions of RBD sequences, the model is exposed to the subtle amino acid changes in the RBD region of the S1 proteins that are present in the GISAID submissions. We reason that the final converged model should be able to generate RBD like sequences that would be very likely to new RBD mutations as long as proper constraints, e.g., virus fitness, increased binding affinity to hACE2, and increased antibody evasion, are satisfied.

We generate RBD variants by performing the following steps. 1) Spike protein sequences are downloaded from the GISAID database, and the sequences in the RBD region are extracted. 2)Training datasets are created from the data processed in step 1. For each VOC, we create a training dataset by using all the RBD sequences from the virus genomes that were submitted before the first appearance of that VOC. 3) The ProtMut model is fine-tuned using the training dataset. 4) A variation probability for each position in the RBD is calculated using the training dataset. 5) The variation probability is used to create masks for each position in the RBD. 6) The variant generation module is used to create amino acids at the masked positions.

5.1.3 High-throughput screening

Once we have generated a large number of mutation sequences, the next step is to simulate the selection pressure faced by viruses through the high-throughput screening. Three screening principles are adopted to conduct the progressive filtering of the generated mutations. Since the main receptor for entering human cells is hACE2, the affinity between the virus RBD and hACE2 is an important indicator for the viral entrance. Therefore, future variants should maintain ideal binding affinity with hACE2. More importantly, various studies have shown that VOCs can escape binding to antibodies. Therefore, we design a model for predicting binding affinity and a model to predict immunity escape of the variants. These two models are bulit with ProtMut as the backbone and are developed based on transfer learning.

As shown in Figure 2b and Figure A1b, we use transfer learning for two types of downstream tasks, which include the prediction of hACE2 binding and antibody binding. The data of quantified changes (refer to Appendix B)of hACE2 binding and antibody binding are first processed to obtain RBD sequences with corresponding amino acid substitutions. Then, the RBD sequences of variants are passed through ProtMut trained on billions of protein sequences to obtain context-aware embeddings, in which the latent pattern at amino-acid-level is captured. With the assistance of the training strategy focused on Local-global feature coupling and sparse hard sample mining (refer to Appendix B), our model outperforms the state-of-the-art baseline models with significant improvements and the hACE2 binding and antibody binding of high-risk variants, including Alpha, Beta, Gamma, Delta, and Omicron, can be correctly predicted. The two downstream tasks are designed to select RBD mutations with better binding affinity to hACE2, and stronger capability of antibody escape.

5.1.4 Simulation of circulating mutations

SAR-CoV-2 is constantly evolving within a host. Due to evolutionary pressures, viruses tend to mutate to acquire stronger fitness, including better binding affinity, and stronger antibody escape capabilities. We simulate

the mutation of SARS-CoV-2 through high-throughput screening and fine-tuning. In each stimulation round, we use the AI models to select those mutations that are predicted to retain ideal binding affinity and stronger antibody escape capabilities. The screened variants will then be used for rounds of fine-tuning of ProtMut. These steps complete the *in silico* mutational simulation of SARS-CoV-2 RBD.

5.2 HPC strategy design

For large-scale distributed AI training, the main goals are to optimize the throughput and speed up network convergence. Pengcheng Cloudbrain-II possesses 4096 pieces of AI processors with 512 server nodes. To efficiently train the language model on such a large cluster, we adopt multiple optimization strategies (Figure 3), reaching a peak performance of 366.8 petaflops in mixed precision.

Operator fusion. We run the training task in graph mode and apply pattern-based operator fusion to accelerate the training in this mode. In this work, we perform fusion on the following operators to optimize the BERT model: (1) We fuse multiple operators for the forward/backward layer normalization operations and perform calculations on multiple NPU cores. (2) We fuse the batch *matmul* operator and the *add* operator. (3) We fuse the All-Reduce operations for all gradients within one Transformer layer into a single operator. These optimizations account for more than 30% of the time consumption.

Operator replacement. Operator replacement refers to the replacement of some operators in a model with new operators that are more amenable to online deployment. In this work, we use fast GeLU in place the of the GeLU operator, which is not very friendly to NPUs. Such operator replacement can improve the model efficiency by 10% while maintaining the accuracy performance.

Operator auto-tuning. AI computing chips are usually composed of computing units, on-chip storage, data transmission, and other modules. The collaboration among these modules will significantly affect the computation patterns of operators. The Auto Tune tool of Ascend uses reinforcement learning and genetic algorithm for tuning particular operators by identifying the optimal tiling policies. We use the Auto Tune tool to optimize the *matmul* operator, which accounts for more than 30% of the time consumption.

Mixed precision. We further improve the speed performance by means of mixed-precision schedules. In dozens of layernorm operators, we schedule a reducing sum operation to the Ascend 910 cube core in FP16 and the other remaining operations to the Ascend 910 vector core in FP32 to avoid computation overflow and achieve higher performance. In addition, the embedding and loss calculations are performed in single precision, and the remaining operators are applied in half precision. The optimizer is implemented in single precision. This mixed-precision implementation greatly reduces the training latency at the cost of potential overflow due to the limited representation range of half precision.

6 HOW PERFORMANCE WAS MEASURED

We perform pretraining of our ProtMut model on Pengcheng Cloudbrain-II with the MindSpore[2] AI computation framework (refer to Appendix C). The system and environment where performance was measured are described in detail in Appendix D. We run tests with 8 NPUs per NPU Pod. The tests are scaled from (1×8) to (512×8) NPUs by powers of 2, and the largest one is assessed on (512×8) NPUs at full-scale. Our model reports timings, including epoch times, mini-batch times, and time-to-solution. We measure the full pretraining time-to-solution, scalability, and peak performance at full-scale. We measure the FLOPS for all precisions by using MindInsight, which is a module of MindSpore. We collect floating-point instructions of relevant flavors (that is, adds, mults, fused multiply adds, and tensor core operations for FP16, FP32, and FP64) and multiply them by corresponding weighting factors, respectively, to transform them into FLOPS counts. The sum of all these values for all precisions yields our overall mixed-precision FLOPS count. In summary, the criteria used to measure the performance of the ProtMut model are defined as follows:

- **Time-to-solution**, defined as the epoch times of strong scaling.
- **Mini-batch size**, defined as the batch size on a single NPU.
- **Peak performance**, defined as $\frac{\text{total FLOPs}}{\text{per step time}}$.

7 PERFORMANCE RESULTS

7.1 Strong scaling performance

The strong scalability of the pretraining process is measured in terms of the epoch times for 1 to 512 nodesof Pengcheng Cloudbrain-II, as shown in Figure 4. For

[2]https://www.mindspore.cn/en

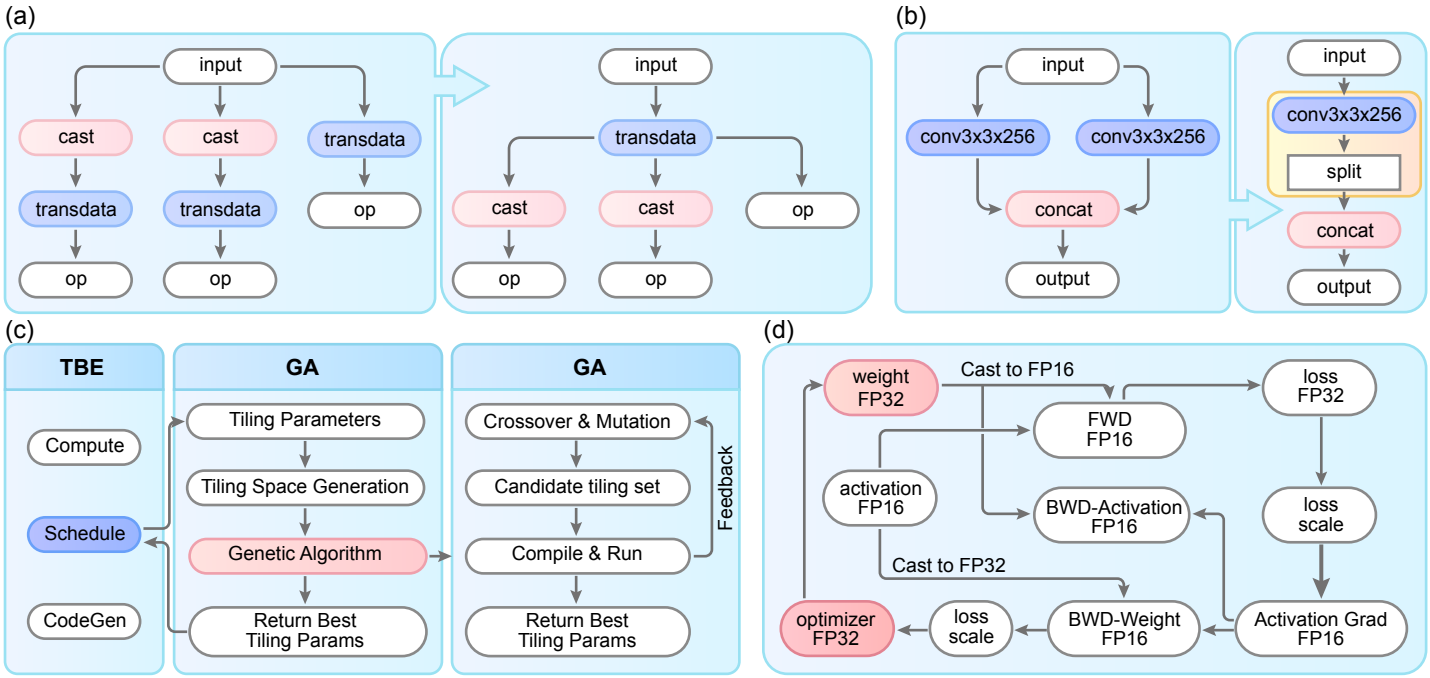

Figure 3: An overview of the employed optimization strategies. a) Operator fusion. To reduce the redundant memory accesses incurred by the successive execution of many small operators, we integrate multiple transdata operators into one transdata operator. b) Operator replacement. We replace two operators with one simplified operator to reduce the computational cost and model size. c) Operator auto-tuning. TBE means Tensor Boosting Engine. GA means Genetic Algorithm. We use a genetic algorithm for tuning particular operators by identifying the optimal tiling policies. A well-designed tiling schedule can fully utilize the computing power of the hardware. d) Mixed precision. All parameters in the model and optimizer are stored in single precision (32-bit), but most of the calculations in this model are performed in half precision (16-bit) to accelerate the training process. This mixed-precision implementation greatly reduces the training latency at the cost of potential overflow due to the limited representation range of half precision.

the strong scaling assessment, the total size of the problem remains the same, i.e., the number of protein sequences used for the protein language model pretraining is kept constant at approximately 408 million. The measured strong scaling, shown as a solid line, almost coincides with the optimal strong scaling, shown as a dotted line, which demonstrates that the strong scaling performance is nearly perfect for 1 to 512 nodes. **With the performance for 1 node as the baseline, the parallel efficiency at 512 nodes is approximately 96.46%, and the speedup reaches about 493.9. In addition, the peak performance reaches 366.81 PFLOPS, and the time-to-solution is 9.1 minutes when scaled to 512 nodes in mixed-precision, which enables rapid deployment and iteration of variant generation models.**

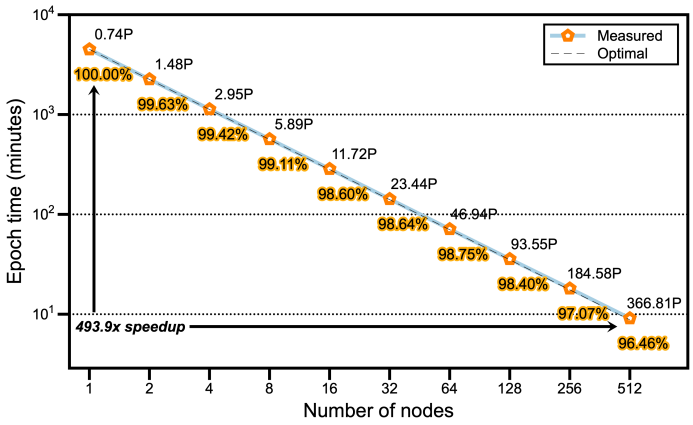

Figure 4: Strong scaling performance of protein language model pretraining for a constant total problem size of approximately 408 million protein sequences. Each data point is labeled with the PFLOPS and parallel efficiency for the corresponding node count. The black dotted line represents the optimal scaling performance for reference.

7.2 Weak scaling performance

As shown in Figure 5, the weak scaling performance of pretraining the protein language model on Pengcheng Cloudbrain-II is also assessed. Unlike the strong scal-

ing case, the problem size per node in the weak scaling test is kept constant at 640 thousand protein sequences. Here, the I/O operations are the saving of checkpoints and train models. Even if the I/O time is included, the degradation in performance at high node is still slight. Specifically, the parallel efficiency for weak scaling from 1 to 512 nodes slightly reduces from 96.73% to 95.57%, and the utilization also remains stable, reducing from 34.99% to 33.54%. In addition, the peak performance can reach 366.86 PFLOPS (34.99% of Peak) when the I/O time is subtracted. In summary, for the pretraining of the protein language model on Pengcheng Cloudbrain-II, the optimized model scales well to the entire supercomputer.

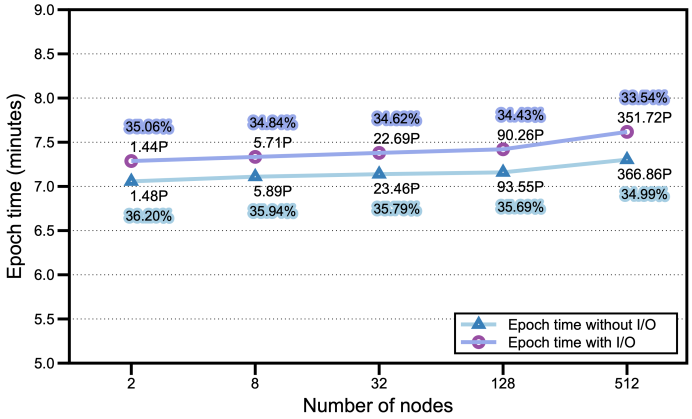

Figure 5: Weak scaling performance of protein language model pretraining for a constant problem size of 640 thousand protein sequences per node. Each data points is labeled with the PFLOPS and utilization for the corresponding node count. Here, the I/O operations include the storage of checkpoints and trained models.

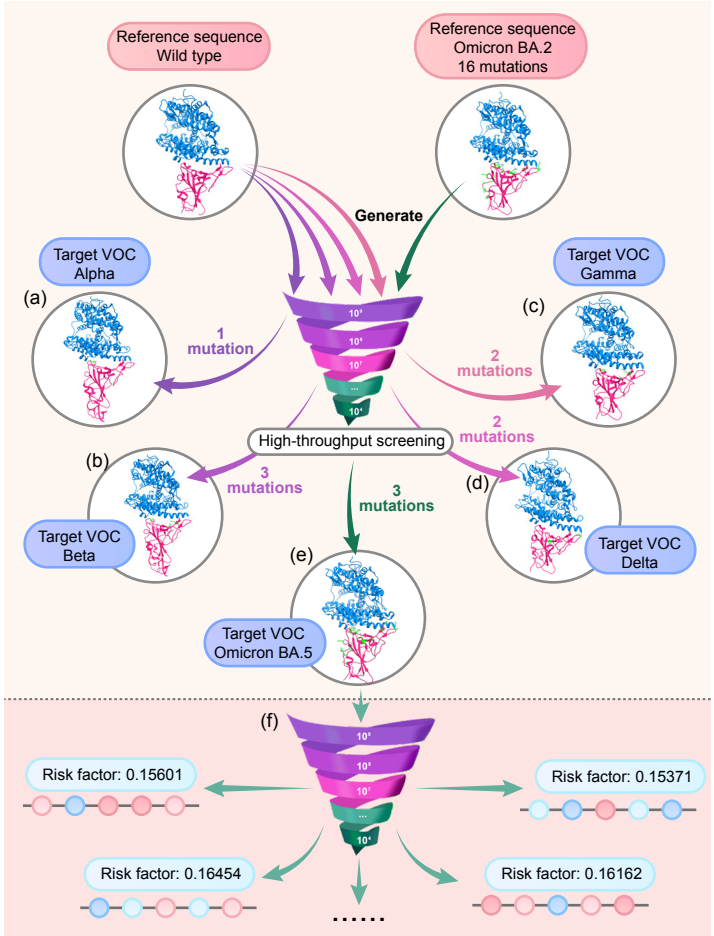

Figure 6: The validation scheme for RBD mutations of VOCs and potential high-risk variants prediction. a), b), c), d) Four VOCs before Omicron, respectively Alpha, Beta, Gamma, and Delta, are simulated from Wild type to themselves. e) Omicron BA.5, a latest subvariant of Omicron, is simulated from Omicron BA.2 to itself. f)Potential high-risk variants prediction. Omicron BA.5 is adopted as the reference sequence, based on the high-throughput screening of hACE2 binding and antibody binding, the risk factor is calculated through transmissibility, which indicates a comprehensive degree of risk.

7.3 *In silico* validation of RBD mutations of VOCs

The VOCs that have emerged to date include B.1.1.7 (Alpha), B.1.351 (Beta), P.1 (Gamma), B.1.617.2 (Delta) and B.1.1.529 (Omicron). Omicron, the currently most widespread VOC, exhibits a several-fold accumulation of variants compared with the first four VOCs, raising the question of whether such variants originally occurred in humans. Some studies have suggested that Omicron may have evolved by accumulating mutations in a mouse host and then jumped back into humans [58, 59]. The possibility of host-jumping makes Omicron significantly different from other VOCs in many ways, such as its mutation profile and the molecular spectrum of its mutations [58, 59]. Therefore, considering the significant difference between the variants before and after the appearance of Omicron, we simulate and verify the RBD mutation process with Omicron as the dividing line.

For SARS-CoV-2 mutation simulation before Omicron, we validate the predictive ability of MCVP by simulating the mutational changes from the Wild Type B.1 to the four VOCs (Alpha, Beta, Gamma, and Delta). According to the pathogenic progression of SARS-CoV-2 [4] based on the data from NextStrain [3], these four VOCs have a parallel evolutionary relationship. Therefore, the starting sequence used to verify the evolutionary route is selected as Wild Type B.1[4]. The sequences

[3] https://nextstrain.org/
[4] EPI_ISL ID: EPI_ISL_402124

used to fine-tune the model are chosen based on when each VOC was first detected. The first detected times and locations of the four VOCs before Omicron are identified via Wikipedia[5]. We segment the data downloaded from GISAID in accordance with the times corresponding to each VOC. For example, Alpha was first reported in September 2020, and we therefore take the data from before September 2020 as the training sequences for fine-tuning ProtMut to predict the emergence of Alpha. Next we adopt the Wild Type as the reference sequence for the mutation generation process. After the RBD mutation generation and high-throughput screening, we check the mutated sites to determine if the RBD of Alpha has appeared in the screened RBD mutations. If it appears, the mutation simulation from Wild Type to Alpha is complete. Otherwise, the filtered RBD mutations are used for iteratively fine-tuning of ProtMut until the RBD of Alpha is generated. Following this simulation method, we have successfully generate the RBDs of the four VOCs (Alpha, Beta, Gamma, and Delta) from the RBD of B.1.

For simulating the evolution of Omicron, we select Omicron BA.2 as the starting point to evolve BA.5 in accordance with the pathogenic progression of SARS-CoV-2 [4]. In this simulation, the sequences of BA.2 and before BA.2 are selected to fine-tune ProtMut, and BA.2 is used as the reference sequence at the time of generation. Through fine-tuning and identification, BA.5 has been generated successfully by our workflow.

Table 1 shows the proportion of remaining variants after each round of screening. Among the above five VOCs, the variants mutated towards Omicron BA.5 retain more than 80% of the proportion in both the hACE binding and antibody escape screening, which indicates that the Omicron sublineages tend to remain stable binding affinity and have stronger antibody escape ability.

Table 1: High throughput screening of various variants

Variant	1st screening*	2st screening**
Alpha	39.8%	2.0%
Beta	13.3%	51.3%
Gamma	45.2%	33.8%
Delta	46.7%	19.1%
Omicron BA.5	90.4%	80.2%

* Proportion after hACE2 binding screening
** Proportion after antibody binding screening

[5]https://en.wikipedia.org/wiki/SARS-CoV-2

7.4 Potential high-risk mutation prediction

By simulating the mutation of the RBD, we have comprehensively demonstrated that MCVP can effectively evolve out the RBDs of the known VOCs. However, the real value of MCVP lies in its ability to predict potential future VOCs, thus assisting targeted drug design and vaccine development.

Omicron has been the dominant variant widely spreading around the world. The phenomenon of intra-VOC evolution has been significant due to the sustained transmission of VOCs, which leading to different descendent lineages. In view of this, a variant tracking system, termed "Omicron subvariants under monitoring", is added to remind us of lineages that need priority attention and monitoring[6]. In this tracking system, BA.5 sublineages (e.g. BF.7, BF.14, BQ.1), BA.2 sublineages (e.g. BA.2.75, BA.2.75.2), and BA.4 sublineage (BA.4.6) need to be focused at present[7]. **In order to demonstrate the potential of MCVP to predict future high-risk variants, we simulate the mutational process of BF.7, BF.14, BQ.1, BA.2.75.2, and BA.4.6. As we expected, we have successfully simulated these variants that WHO reminds public health authorities around the world to give priority to.**

More importantly, we take the latest sublineage of Omicron, i.e. BA.5, as the reference sequence, then generate billions of variants in each round and conduct subsequent high-throughput screening. After evaluation of binding affinity and antibody escape capability, we use the screened sequences to fine-tune ProtMut. After several rounds of iterations, we select a number of potential RBD mutations with higher risk that maintain a stable binding affinity with hACE2 and a high antibody escape capability. At this stage, to better evaluate potential VOCs, we adopt PyR_0 [60], a hierarchical Bayesian multinomial logistic regression model, to score the generated variants for transmissibility. We calculate the comprehensive risk factor based on the high-throughput screening and the transmissibility scores of variant. A variant whose risk factor is greater than 0 may have greater risk than Omicron BA.5, and a variant whose risk factor is less than 0 may have less risk. **As a result, billions of variants can be evaluated quickly.**

[6]https://www.who.int/activities/tracking-SARS-CoV-2-variants

[7]https://www.cbsnews.com/news/covid-variants-ba46-bf7-ba275-rise-cdc-tracking/

8 Implications

8.1 AI models can successfully generate and identify almost all VOCs

In our experiments, using genomic data submitted before the appearance of each VOCs, we successfully generate and identify all VOCs except Omicron. Given the original Omicron spike sequences, we could also generate the Omicron subvariants that are currently the dominant viral variants throughout the world.

During the iterative mutation generation process, the AI models can prioritize mutations based on their predicted fitness, binding affinity prediction and antibody escape, three key factors for viral infectivity. Due to their combinatorial nature, it is impossible to experimentally measure the binding affinity changes among all possible RBD mutations ($20^{201} = 3.213876 \times 10^{261}$) and hACE2 or antibodies. Therefore, under the assumption that the DMS measurements of RBD single mutations might provide reasonable constraints for the RBD to hACE2/antibody binding affinity spaces, we approximate these binding affinity spaces using DL models for prediction of the binding affinities among multiple RBD mutations and hACE2 or antibodies. These AI models are key innovations of the whole workflow.

There are several hypotheses regarding the origin of Omicron, including the possibility of spillover from mice [58]. The fact that our workflow could not generate Omicron despite more than 20 rounds of iteration implies that the mutational features of Omicron are very different from those of other VOCs, since all other VOCs were found after a few rounds of generation.

8.2 The simulation of SARS-CoV-2 spike mutation is an HPC application

The strategy we used to simulate SARS-CoV-2 spike mutation is dependent on the availability of large-scale genome data (more than 13 million viral genomes as provided by the GISAID database) and a large protein language generation model.

Recent progress in Transformer-based models has enabled the implementation of protein language models capable of generating de novo protein sequences following the principles of natural ones [61]. Inspired by these successes, we pretrain a BERT-like model to learn from millions of viral spike proteins. Our mutation generation workflow heavily relies on the Pengcheng Cloudbrain-II: first, to train the protein language model; second, to iteratively generate new mutations; and third, to evaluate the mutations based on AI predictors of: 1)the binding affinity between RBD and hACE2, 2)the antibody escape capability. All of the processing steps require an HPC facility, as billions of RBD mutations must be generated in each round and evaluated accordingly.

8.3 Simulating coronavirus evolution is a new challenge for HPC

The COVID-19 pandemic, caused by SARS-CoV-2, is a stark reminder that coronaviruses remain a major threat to humanity. It is crucial to study the evolution of Coronaviruses to be better prepared for the next pandemic.

SARS-CoV-2 has become the most sequenced virus ever in history, with 13 million SARS-CoV-2 genomes deposited in the GISAID database. The efficiency of simulating these extremely large numbers of closely related genomes to recreate potential histories of past and future virus evolution presents a new challenge for HPC. As a proof of concept, in this study, we have initialized a small step toward elucidating the evolution of SARS-CoV-2 VOCs by using only RBD sequences of the SARS-CoV-2 S1 protein. By using the whole genomes of SARS-CoV-2 in the future, plus other coronavirus genomes, we will be able to perform more reliable simulations to study the evolution of coronaviruses in general and the dynamics of viral transmission across animal species. Meeting the computational requirements of such simulations will demand some of the finest HPC systems built to date.

8.4 SARS-CoV-2 mutation is a serious threat

It has been estimated that an infected person could carry 10^9 to 10^{12} SARS-CoV-2 virions [62]. Since the initial outbreak of COVID-19, there have been 612 million infections as of September 2022 [8]. The potential mutation space for SARS-CoV-2 is thus approximately 6×10^{17} to 10^{20}. The experimentally deduced spontaneous mutation rate of SARS-CoV-2 is $1.3 \times 10^{-6} \pm 0.2 \times 10^{-6}$ per_base per_infection cycle [63], which is heterogeneous throughout the genome. Taking all these numbers together, it is not too difficult to conclude that every single base mutation is being generated de-novo and transmitted to a new host every day [62]. It is therefor extremely important to be able to simulate the viral mutation process and rapidly identify potential VOCs, which is essentially what we have demonstrated in this work through the state-of-the-art AI technology combined with the cutting-edge HPC hardware - the Pengcheng Cloudbrain-II. Any successful predictions of future VOCs of SARS-CoV-2 is not just good scientific research but can prevent unnecessary deaths.

[8] https://covid19.who.int/

REFERENCE

[1] M. M. Lamers and B. L. Haagmans, "Sars-cov-2 pathogenesis," *Nature Reviews Microbiology*, pp. 1–15, 2022.

[2] C. S. G. of the International *et al.*, "The species severe acute respiratory syndrome-related coronavirus: classifying 2019-ncov and naming it sars-cov-2," *Nature microbiology*, vol. 5, no. 4, p. 536, 2020.

[3] P. Zhou, X.-L. Yang, X.-G. Wang, B. Hu, L. Zhang, W. Zhang, H.-R. Si, Y. Zhu, B. Li, C.-L. Huang *et al.*, "A pneumonia outbreak associated with a new coronavirus of probable bat origin," *nature*, vol. 579, no. 7798, pp. 270–273, 2020.

[4] E. Callaway *et al.*, "Are covid surges becoming more predictable? new omicron variants offer a hint," *Nature*, vol. 605, no. 7909, pp. 204–206, 2022.

[5] Y. Yin and R. G. Wunderink, "Mers, sars and other coronaviruses as causes of pneumonia," *Respirology*, vol. 23, no. 2, pp. 130–137, 2018.

[6] C. Drosten, S. Günther, W. Preiser, S. Van Der Werf, H.-R. Brodt, S. Becker, H. Rabenau, M. Panning, L. Kolesnikova, R. A. Fouchier *et al.*, "Identification of a novel coronavirus in patients with severe acute respiratory syndrome," *New England journal of medicine*, vol. 348, no. 20, pp. 1967–1976, 2003.

[7] A. M. Zaki, S. Van Boheemen, T. M. Bestebroer, A. D. Osterhaus, and R. A. Fouchier, "Isolation of a novel coronavirus from a man with pneumonia in saudi arabia," *New England Journal of Medicine*, vol. 367, no. 19, pp. 1814–1820, 2012.

[8] S. Su, G. Wong, W. Shi, J. Liu, A. C. Lai, J. Zhou, W. Liu, Y. Bi, and G. F. Gao, "Epidemiology, genetic recombination, and pathogenesis of coronaviruses," *Trends in microbiology*, vol. 24, no. 6, pp. 490–502, 2016.

[9] R. Lu, X. Zhao, J. Li, P. Niu, B. Yang, H. Wu, W. Wang, H. Song, B. Huang, N. Zhu *et al.*, "Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding," *The lancet*, vol. 395, no. 10224, pp. 565–574, 2020.

[10] A. C. Walls, Y.-J. Park, M. A. Tortorici, A. Wall, A. T. McGuire, and D. Veesler, "Structure, function, and antigenicity of the sars-cov-2 spike glycoprotein," *Cell*, vol. 181, no. 2, pp. 281–292, 2020.

[11] D. Wrapp, N. Wang, K. S. Corbett, J. A. Goldsmith, C.-L. Hsieh, O. Abiona, B. S. Graham, and J. S. McLellan, "Cryo-em structure of the 2019-ncov spike in the prefusion conformation," *Science*, vol. 367, no. 6483, pp. 1260–1263, 2020.

[12] C. Wu, Y. Liu, Y. Yang, P. Zhang, W. Zhong, Y. Wang, Q. Wang, Y. Xu, M. Li, X. Li *et al.*, "Analysis of therapeutic targets for sars-cov-2 and discovery of potential drugs by computational methods," *Acta Pharmaceutica Sinica B*, vol. 10, no. 5, pp. 766–788, 2020.

[13] Y. Wan, J. Shang, R. Graham, R. S. Baric, and F. Li, "Receptor recognition by the novel coronavirus from wuhan: an analysis based on decade-long structural studies of sars coronavirus," *Journal of virology*, vol. 94, no. 7, pp. e00 127–20, 2020.

[14] M. Hoffmann, H. Kleine-Weber, S. Schroeder, N. Krüger, T. Herrler, S. Erichsen, T. S. Schiergens, G. Herrler, N.-H. Wu, A. Nitsche *et al.*, "Sars-cov-2 cell entry depends on ace2 and tmprss2 and is blocked by a clinically proven protease inhibitor," *cell*, vol. 181, no. 2, pp. 271–280, 2020.

[15] W. Li, M. J. Moore, N. Vasilieva, J. Sui, S. K. Wong, M. A. Berne, M. Somasundaran, J. L. Sullivan, K. Luzuriaga, T. C. Greenough *et al.*, "Angiotensin-converting enzyme 2 is a functional receptor for the sars coronavirus," *Nature*, vol. 426, no. 6965, pp. 450–454, 2003.

[16] S. Belouzard, V. C. Chu, and G. R. Whittaker, "Activation of the sars coronavirus spike protein via sequential proteolytic cleavage at two distinct sites," *Proceedings of the National Academy of Sciences*, vol. 106, no. 14, pp. 5871–5876, 2009.

[17] G. Lu, Q. Wang, and G. F. Gao, "Bat-to-human: spike features determining 'host jump'of coronaviruses sars-cov, mers-cov, and beyond," *Trends in microbiology*, vol. 23, no. 8, pp. 468–478, 2015.

[18] X. Chi, R. Yan, J. Zhang, G. Zhang, Y. Zhang, M. Hao, Z. Zhang, P. Fan, Y. Dong, Y. Yang *et al.*, "A neutralizing human antibody binds to the n-terminal domain of the spike protein of sars-cov-2," *Science*, vol. 369, no. 6504, pp. 650–655, 2020.

[19] T. M. Gallagher and M. J. Buchmeier, "Coronavirus spike proteins in viral entry and pathogenesis," *Virology*, vol. 279, no. 2, pp. 371–374, 2001.

[20] G. Simmons, P. Zmora, S. Gierer, A. Heurich, and S. Pöhlmann, "Proteolytic activation of the sars-coronavirus spike protein: cutting enzymes at the

cutting edge of antiviral research," *Antiviral research*, vol. 100, no. 3, pp. 605–614, 2013.

[21] S. Duffy, "Why are rna virus mutation rates so damn high?" *PLoS biology*, vol. 16, no. 8, p. e3000003, 2018.

[22] F. Obermeyer, M. Jankowiak, N. Barkas, S. F. Schaffner, J. D. Pyle, L. Yurkovetskiy, M. Bosso, D. J. Park, M. Babadi, B. L. MacInnis *et al.*, "Analysis of 6.4 million sars-cov-2 genomes identifies mutations associated with fitness," *medRxiv*.

[23] Y. Shu and J. McCauley, "Gisaid: Global initiative on sharing all influenza data–from vision to reality," *Eurosurveillance*, vol. 22, no. 13, p. 30494, 2017.

[24] R. G. Beiko and R. L. Charlebois, "A simulation test bed for hypotheses of genome evolution," *Bioinformatics*, vol. 23, no. 7, pp. 825–831, 2007.

[25] R. R. Hudson, "Generating samples under a wright–fisher neutral model of genetic variation," *Bioinformatics*, vol. 18, no. 2, pp. 337–338, 2002.

[26] G. Laval and L. Excoffier, "Simcoal 2.0: a program to simulate genomic diversity over large recombining regions in a subdivided population with a complex history," *Bioinformatics*, vol. 20, no. 15, pp. 2485–2487, 2004.

[27] G. Ewing and J. Hermisson, "Msms: a coalescent simulation program including recombination, demographic structure and selection at a single locus," *Bioinformatics*, vol. 26, no. 16, pp. 2064–2065, 2010.

[28] A. Rambaut and N. C. Grass, "Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees," *Bioinformatics*, vol. 13, no. 3, pp. 235–238, 1997.

[29] W. Fletcher and Z. Yang, "Indelible: a flexible simulator of biological sequence evolution," *Molecular biology and evolution*, vol. 26, no. 8, pp. 1879–1888, 2009.

[30] B. Sipos, T. Massingham, G. E. Jordan, and N. Goldman, "Phylosim-monte carlo simulation of sequence evolution in the r statistical computing environment," *BMC bioinformatics*, vol. 12, no. 1, pp. 1–6, 2011.

[31] N. De Maio, W. Boulton, L. Weilguny, C. R. Walker, Y. Turakhia, R. Corbett-Detig, and N. Goldman, "phastsim: efficient simulation of sequence evolution for pandemic-scale datasets," *PLoS computational biology*, vol. 18, no. 4, p. e1010056, 2022.

[32] V. Shchur, V. Spirin, V. Pokrovsky, D. Sirotkin, E. Burovski, N. De Maio, and R. Corbett-Detig, "Vgsim: scalable viral genealogy simulator for global pandemic," *medRxiv*, 2021.

[33] J. Chen, R. Wang, M. Wang, and G.-W. Wei, "Mutations strengthened sars-cov-2 infectivity," *Journal of molecular biology*, vol. 432, no. 19, pp. 5212–5226, 2020.

[34] T. Mohamed, S. Sayed, A. Salah, and E. H. Houssein, "Next generation sequence prediction intelligent system for sars-cov-2 using deep learning neural network," in *2021 17th International Computer Engineering Conference (ICENCO)*. IEEE, 2021, pp. 88–93.

[35] K. Ito, C. Piantham, and H. Nishiura, "Predicted dominance of variant delta of sars-cov-2 before tokyo olympic games, japan, july 2021," *Eurosurveillance*, vol. 26, no. 27, p. 2100570, 2021.

[36] F. Pucci and M. Rooman, "Prediction and evolution of the molecular fitness of sars-cov-2 variants: Introducing spikepro," *Viruses*, vol. 13, no. 5, p. 935, 2021.

[37] J. Li, Y.-N. Wu, S. Zhang, X.-P. Kang, and T. Jiang, "Deep learning based on biologically interpretable genome representation predicts two types of human adaptation of sars-cov-2 variants," *Briefings in Bioinformatics*, vol. 23, no. 3, p. bbac036, 2022.

[38] D. Ofer, N. Brandes, and M. Linial, "The language of proteins: Nlp, machine learning & protein sequences," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 1750–1758, 2021.

[39] E. N. Trifonov, "The origin of the genetic code and of the earliest oligopeptides," *Research in microbiology*, vol. 160, no. 7, pp. 481–486, 2009.

[40] B. J. Strait and T. G. Dewey, "The shannon information entropy of protein sequences," *Biophysical journal*, vol. 71, no. 1, pp. 148–155, 1996.

[41] L. Yu, D. K. Tanwar, E. D. S. Penha, Y. I. Wolf, E. V. Koonin, and M. K. Basu, "Grammar of protein domain architectures," *Proceedings of the National Academy of Sciences*, vol. 116, no. 9, pp. 3636–3645, 2019.

[42] Y. Wu, S. Xu, S.-T. Yau, and Y. Wu, "Phylotransformer: A discriminative model for mutation prediction based on a multi-head self-attention mechanism," *arXiv preprint arXiv:2111.01969*, 2021.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[44] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[45] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[46] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[47] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[48] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, 2021.

[49] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger *et al.*, "Prottrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing," *arXiv preprint arXiv:2007.06225*, 2020.

[50] X. He, F. Xue, X. Ren, and Y. You, "Large-scale deep learning optimizations: A comprehensive survey," *arXiv preprint arXiv:2111.00856*, 2021.

[51] Z. Li, E. Wallace, S. Shen, K. Lin, K. Keutzer, D. Klein, and J. Gonzalez, "Train big, then compress: Rethinking model size for efficient training and inference of transformers," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5958–5968.

[52] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[53] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *International Conference on Learning Representations, ICLR*, 2017.

[54] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[55] K. Osawa, Y. Tsuji, Y. Ueno, A. Naruse, C. Foo, and R. Yokota, "Scalable and practical natural gradient for large-scale deep learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 404–415, 2022.

[56] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, "Uniref: comprehensive and non-redundant uniprot reference clusters," *Bioinformatics*, vol. 23, no. 10, pp. 1282–1288, 2007.

[57] T. N. Starr, A. J. Greaney, S. K. Hilton, D. Ellis, K. H. Crawford, A. S. Dingens, M. J. Navarro, J. E. Bowen, M. A. Tortorici, A. C. Walls *et al.*, "Deep mutational scanning of sars-cov-2 receptor binding domain reveals constraints on folding and ace2 binding," *cell*, vol. 182, no. 5, pp. 1295–1310, 2020.

[58] C. Wei, K.-J. Shan, W. Wang, S. Zhang, Q. Huan, and W. Qian, "Evidence for a mouse origin of the sars-cov-2 omicron variant," *Journal of genetics and genomics*, vol. 48, no. 12, pp. 1111–1121, 2021.

[59] Y. Sun, W. Lin, W. Dong, and J. Xu, "Origin and evolutionary analysis of the sars-cov-2 omicron variant," *Journal of Biosafety and Biosecurity*, vol. 4, no. 1, pp. 33–37, 2022.

[60] F. Obermeyer, M. Jankowiak, N. Barkas, S. F. Schaffner, J. D. Pyle, L. Yurkovetskiy, M. Bosso, D. J. Park, M. Babadi, B. L. MacInnis *et al.*, "Analysis of 6.4 million sars-cov-2 genomes identifies mutations associated with fitness," *Science*, vol. 376, no. 6599, pp. 1327–1332, 2022.

[61] N. Ferruz, S. Schmidt, and B. Höcker, "Protgpt2 is a deep unsupervised language model for protein

design," *Nature communications*, vol. 13, no. 1, pp. 1–10, 2022.

[62] R. Sender, Y. M. Bar-On, S. Gleizer, B. Bernshtein, A. Flamholz, R. Phillips, and R. Milo, "The total number and mass of sars-cov-2 virions," *Proceedings of the National Academy of Sciences*, vol. 118, no. 25, p. e2024815118, 2021.

[63] M. Amicone, V. Borges, M. J. Alves, J. Isidro, L. Zé-Zé, S. Duarte, L. Vieira, R. Guiomar, J. P. Gomes, and I. Gordo, "Mutation rate of sars-cov-2 and emergence of mutators during experimental evolution," *Evolution, medicine, and public health*, vol. 10, no. 1, pp. 142–155, 2022.

[64] M. Steinegger, M. Mirdita, and J. Söding, "Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold," *Nature methods*, vol. 16, no. 7, pp. 603–606, 2019.

[65] M. Steinegger and J. Söding, "Clustering huge protein sequence sets in linear time," *Nature communications*, vol. 9, no. 1, pp. 1–8, 2018.

[66] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, "Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2015.

[67] R. Leinonen, F. G. Diez, D. Binns, W. Fleischmann, R. Lopez, and R. Apweiler, "Uniprot archive," *Bioinformatics*, vol. 20, no. 17, pp. 3236–3237, 2004.

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[69] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," *arXiv preprint arXiv:1904.00962*, 2019.

[70] A. J. Greaney, T. N. Starr, P. Gilchuk, S. J. Zost, E. Binshtein, A. N. Loes, S. K. Hilton, J. Huddleston, R. Eguia, K. H. Crawford *et al.*, "Complete mapping of mutations to the sars-cov-2 spike receptor-binding domain that escape antibody recognition," *Cell host & microbe*, vol. 29, no. 1, pp. 44–57, 2021.

[71] H. Liao, J. Tu, J. Xia, and X. Zhou, "DaVinci: A scalable architecture for neural network computing," in *IEEE Hot Chips 31 Symposium (HCS)*, 2019, pp. 1–44.

Appendix

A Pretrained protein language model

The data used to pretrain our protein language model ProtMut and the model details shown in Figure A1 are as follows:

Data. There are many existing protein databases such as BFD [64, 65], UniRef50 [66], UiRef90 [66], and UniRef100 [66]. UniRef is an abbreviation for UniProt Reference Clusters, consisting of clustered sets of protein sequences from the UniProt Knowledgebase (UniProtKB) [67]. The UniRef100 database combines identical sequences and subfragments from any source organism into a single UniRef entry. UniRef90 and UniRef50 are built by clustering UniRef100 sequences at the 90% or 50% sequence identity levels. Since the number of mutations accounts for approximately 1% $\sim$ 10% of the RBD region, we take UniRef90, which contains 144 million proteins, as the training set.

We perform the following data processing procedure: 1) we remove all ambiguous amino acids, 2) we designated every 256 amino acids from the protein sequences as one training sequence, and 3) we discard protein sequences with a length of less than 10. As a result, a total of 408 million training sequences are used to pretrain the protein language model. A tokenization process is employed to transform any given protein sequences into a model-recognizable format by building a vocabulary for the model, which consists of a mapping between protein sequences and unique integer ids. We assign an integer ID for each amino acid, plus the MASK token, resulting in a vocabulary size of 22.

Model details. Our model is based on BERT [43]. By learning to reconstruct the corrupted tokens in the training data, BERT becomes able to extract embedding vectors for protein sequences. The architecture of BERT is based on the Transformer model [68]. The self-attention mechanism in Transformer relies on an order-independent operation. Thus, BERT requires explicit positional encoding. Here, we use the learned positional encoding. To a large extent, we transfer the configuration of the BERT-large model from natural language to protein sequences: the number of layers is 24, the hidden layer size is 1024, the intermediate size of the MLP block is 4096, and the number of attention heads

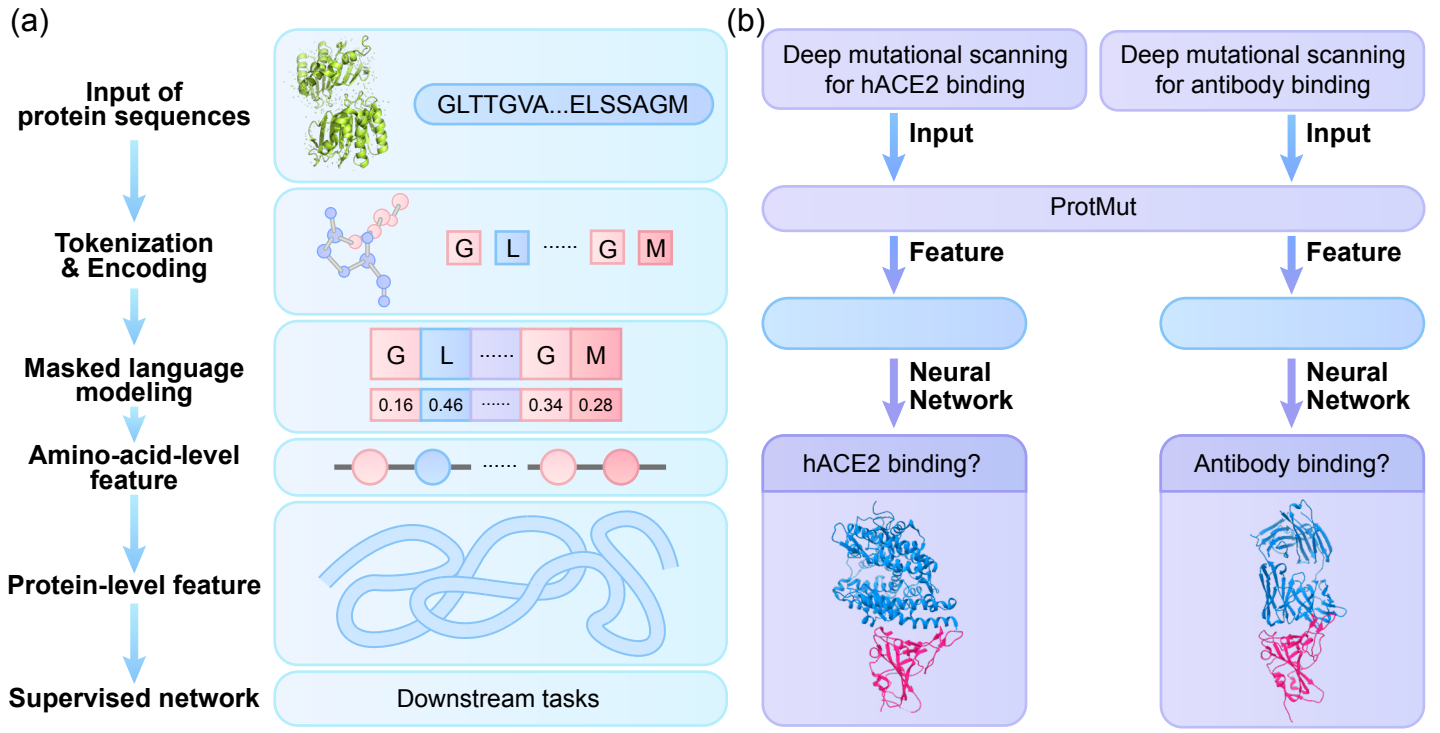

Figure A1: a) A protein language model, called ProtMut here, is pre-trained on the massive protein sequences. b) High-throughput screening design. Two deep learning models are developed based on ProtMut to achieve the identification of high-risk variants.

is 16. Thus, the total number of parameters is 340 million. During pretraining, random tokens of each input protein sequence are masked, and the model is trained to predict the identity of the masked tokens based on the surrounding context. The masking probability of the input amino acids is 15%. We treat the prediction of each masked token as a classification problem and employ the Cross-Entropy loss function to optimize the model. The optimizer used is Lamb [69] which enables the use of very large batch size. The weight decay is 0.01. The learning rate ranges from 1e-4 to 1e-3 for various nodes. We used the cosine decay for the learning rate schedule.

B Transfer learning models for high-throughput screening

As shown in Figure A1b,for the hACE2 binding prediction model, the quantified changes of binding affinity of RBD and hACE2 in the deep mutagenesis study by Starr et al [57] are adopted. And for the antibody binding prediction model, the antibody escape fractions for each mutation computed by deep-sequencing counts are used [70]. To fuse local-level relationships between adjacent amino acids and long-range relationships between distant amino acids, the module of feature refinement including CNN and self-attention is designed to combine the local feature and global feature. However, the above deep mutational scanning data is extremely class-imbalanced. Considering the original Mean Squared Error (MSE) or Binary Cross-Entropy (BCE) loss function can't handle sparse hard samples, we design multi-task focal loss to alleviate the adverse effects of class-imbalanced and improve the generalization of our model. The fused comprehensive features are used for training multilayer perceptrons (MLP) for different downstream tasks.

C Applications used to measure performance

In this work, we use the UniRef90 protein dataset to pretrain the ProtMut model. After that, we fine-tune ProtMut with RBD sequences truncated from the Spike protein sequences downloaded from GISAID. Finally, we iteratively fine-tune the model with generated RBD mutations. All models are implemented in source code using the MindSpore DL library. Since the pretraining of the protein language model is the most computationally expensive phase of this study, it is the focus of the performance analysis. For protein language modeling, the architecture is based on BERT, with approximately 340 million learnable parameters. The architecture hyper-

parameters of ProtMut are shown in Table C1. Hybrid parallelism (data parallelism and model parallelism) is adopted to scale out to the whole cluster and improve the training performance.

Table C1: Hyperparameters of ProtMut Architecture

Hyperparameter	Value
embedding_size	1024
layers	24
sequence length	256
attention head nums	16
ffn_hidden_size	4096

D System and environment where performance was measured

We run our tests on Pengcheng Cloudbrain-II, which contains 4096 Ascend 910 [71] AI processors (NPUs) and 2048 Kunpeng 920 processors. FigureD1 shows the structure diagram of Pengcheng Cloudbrain-II AI computing system. Each NPU provides 256 TFLOPS of half-precision (FP16) peak computing power, 512 TFLOPS of integer precision (INT8) peak computing power, has a maximum power consumption of 350 W, and 32 GB of high-bandwidth storage. The general computing power is provided by Kunpeng 920 processors based on the ARM architecture, each of which provides a single-precision (FP32) peak computing power of 0.98 TOPS and a double-precision (FP64) peak computing power of 0.49 TOPS. As the entire AI computing system contains a total of 4096 NPUs and 2048 Kunpeng 920 processors, the total peak AI computing power is 1048 PFLOPS in half-precision or 2048 TOPS in integer-precision, and the total peak general computing power is 2007 TFLOPS in single-precision or 1003.5 TOPS in double-precision.

Pengcheng Cloudbrain-II runs the openEuler operating system. The DL library we used here is MindSpore, which is an open source DL training/inference framework that can be used in mobile, edge and cloud scenarios. The host compiler is gcc 7.3.0. The programming language is Python 3.7.5.

E Single NPU Pod configuration

Larger mini-batch sizes will enable higher computational efficiency, but could harm the learning effect. To determine the appropriate mini-batch size per NPU, we carry out ablation experiments on an NPU Pod ($8 \times 8 = 64$ NPUs) of Pengcheng Cloudbrain-II. The results in Figure E1, including mini-batch time and FLOPS utilization rate, demonstrate that 80 is the optimal setting, with over 36% utilization rate. Hence, in all subsequent scaling tests, a mini-batch size configuration of 80 per NPU is used.

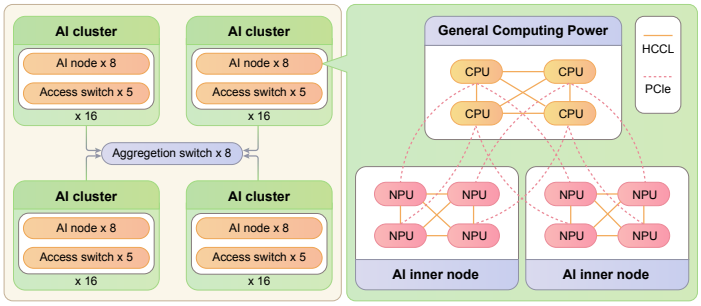

Figure D1: AI computing system architecture. The Pengcheng Cloudbrain-II AI supercomputer includes 4 AI computing clusters. Each cluster contains 16 NPU Pods, which are interconnected through high-speed through aggregation switches. Each NPU Pod contains 8 AI computing nodes, which are interconnected through high-speed access switches. Each AI computing node contains 8 AI processors (NPUs) and 4 Kunpeng 920 processors. Every 4 NPUs form an inner-node cluster, and communicate with each other through PCIe 4.0.

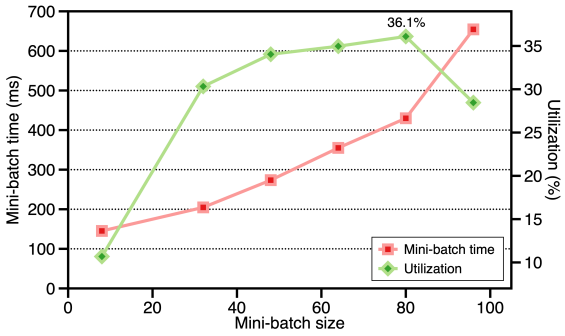

Figure E1: Ablation experiments on an NPU Pod. The trends of mini-batch time and FLOPS utilization rate with different mini-batch sizes are shown in red and green, respectively.