bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Incorporating Pre-training Paradigm for Antibody Sequence-Structure Co-design

Kaiyuan Gao[1]*, **Lijun Wu**[2]†, **Jinhua Zhu**[3], **Tianbo Peng**[4], **Yingce Xia**[2], **Liang He**[2], **Shufang Xie**[2], **Tao Qin**[2], **Haiguang Liu**[2], **Kun He**[1], **Tie-Yan Liu**[2]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology;
[2]Microsoft Research AI4Science;
[3]CAS Key Laboratory of GIPAS, University of Science and Technology of China;
[4]School of Life Sciences and Biomedical Pioneering Innovation Center, Peking University;
{im_kai, brooklet60}@hust.edu.cn
{teslazhu}@mail.ustc.edu.cn, ptbian@pku.edu.cn
{lijuwu, yinxia, lihe, shufxi, taoqin, haiguangliu, tyliu}@microsoft.com

Abstract

Antibodies are versatile proteins that can bind to pathogens and provide effective protection for human body. Recently, deep learning-based computational antibody design has attracted popular attention since it automatically mines the antibody patterns from data that could be complementary to human experiences. However, the computational methods heavily rely on the high-quality antibody structure data, which is quite limited. Besides, the complementarity-determining region (CDR), which is the key component of an antibody that determines the specificity and binding affinity, is highly variable and hard to predict. Therefore, data limitation issue further raises the difficulty of CDR generation for antibodies. Fortunately, there exists a large amount of sequence data of antibodies that can help model the CDR and alleviate the reliance on structured data. By witnessing the success of pre-training models for protein modeling, in this paper, we develop an antibody pre-trained language model and incorporate it into the (antigen-specific) antibody design model in a systemic way. Specifically, we first pre-train an antibody language model based on the sequence data, then propose a one-shot way for sequence and structure generation of CDR to avoid the heavy cost and error propagation from an autoregressive manner, and finally leverage the pre-trained antibody model for the antigen-specific antibody generation model with some carefully designed modules. Through various experiments, we show that our method achieves superior performance over previous baselines on different tasks, such as sequence and structure generation, antigen-binding CDR-H3 design.

1 Introduction

Antibodies are Y-shaped proteins (Figure 1 for the overall structure of an antibody) and they are crucial biological elements in human immune system as therapeutics targeting various pathogens, treating cancer, infectious diseases and so on [41]. They have several characteristics. First, the antibodies have strong specificity towards the effectiveness [36]. Most antibodies are monoclonal that each kind of antibody usually binds to a unique type of protein (antigen). Second, the binding areas of antibodies are mainly determined by complementarity-determining regions (CDR), while the CDRs are highly variable with free loop and other unstructured shapes, especially the CDRs on

*This work is conducted at Microsoft Research AI4Science.

†Corresponding author.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

the heavy chain [52]. Therefore, the crucial problem of antibody design mainly focuses on how to identify and design novel CDRs that can effectively and stably bind to the specific antigen.

Recently, computational methods [11, 13, 28] have been explored to automatically create the CDR sequences with desired properties, e.g., high binding affinity. Especially, deep learning has demonstrated its great potential for the antibody design, such as deep generative models [39], graph neural networks [42]. Previously, people focus more on generating the CDR sequences only [30, 56, 3, 43]. However, co-designing the sequences with their 3D structures is a more promising choice because of its realistic/practical value and it becomes a recent trend. For example, [16] and [17] both utilize the deep graph neural networks to model the antibody generation from sequence and structure. Though huge progress has been made, there still exists several key challenges. (1) The 3D structure data of antibodies in existing dataset is in limited amount. For example, SAbDab [8], the most widely used dataset for antibody design, with daily collecting, it still only contains thousands of antibody structures. As for the antigen-antibody complexes, which are the key ingredients to model the interaction between antibodies and antigen, the number is even more limited. However, the antibody space of a single individual is estimated to be at least 10^{12} [40], which is far beyond the volume of existing database. Also, compared with the common applications that achieved big success, this kind of data scale is far from enough for training deep learning models Besides, due to the highly variable property of the CDRs, the limited existing structures further hinders the learning ability for deep models to accurately predict the CDR structures. (2) The current approaches for antibody design usually adopt the autoregressive generation manner [16], which is to predict the amino acid type and the corresponding structure coordinates one by one. The drawbacks of such a way are obvious. On the one hand, the step-by-step amino acid generation suffers from low efficiency that it will take multiple iterations (the length of the CDR) to complete the CDR design. On the other hand, it is also well known that this autoregressive way faces the error accumulation problem [35, 53], where the error from the previous step will propagate to next step during generation so that the accuracy is influenced. These issues lead to unsatisfactory performance on the sequence and structure co-design. For instance, the amino acid recovery (AAR) rate is only about 30% for the CDR-H3 (the third CDR on the heavy chain), which has huge space for improvement.

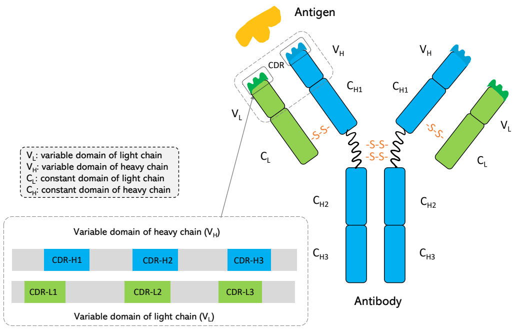

Antigen
CDR
V_H
V_H
C_{H1}
C_{H1}
V_L
V_L
-S-S-
-S-S-
-S-S-
-S-S-
C_L
C_L
V_L: variable domain of light chain
V_H: variable domain of heavy chain
C_L: constant domain of light chain
C_H: constant domain of heavy chain
C_{H2}
C_{H2}
Variable domain of heavy chain (V_H)
CDR-H1
CDR-H2
CDR-H3
C_{H3}
C_{H3}
CDR-L1
CDR-L2
CDR-L3
Variable domain of light chain (V_L)
Antibody

Figure 1: The Y-shape of an antibody. The CDR-Hn are the variable CDRs that belong to the heavy chain, which are the most important parts for binding. Below is the sequential format of the variable domains of two chains.

To address above challenges, in this paper, we propose several strategies that can help from different perspectives. (a) Although the structured antibody data is in limited scale, there are millions of antibody sequence data that we can leverage. It is widely acknowledged that the sequence of the protein can determine its structure [23, 10]. Sequence pre-training has demonstrated its power in various domains [20, 29, 27, 58], and the protein sequence pre-training also demonstrates its ability for protein structure prediction [55, 7]. Therefore, we utilize the large amount of the sequence data for antibody pre-training. By doing so, the antibody representations can be greatly enhanced and thus alleviate the problem brought by the lack of structured data. (b) We formulate the antibody generation process in a one-shot way rather than the autoregressive manner. That is, we simultaneously generate all the amino acids of CDR at once, and the structure of the generated amino acids are also updated at one time. In this way, the error propagation problem [35] between different steps of the autoregressive way is avoided. To ensure the performance, we also refine the whole generated sequences and structures of CDR with several iterations like Alphafold2 [18]. (c) We incorporate the pre-trained antibody language model into the antibody design model in a careful way. Instead of only taking the prediction of the pre-trained model as initialization for the antibody design model, we introduce two integration ways to make the best use of the pre-trained model. We use prompt tuning [24, 26] strategy to better finetune the pre-trained model, which is supposed to not only keep the ability of the pre-trained model but also provide the transferable knowledge for downstream antibody design. Besides, we systematically fuse the intermediate representations of the two models and then feed for

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

sequence and structure prediction of CDR. Through above designed strategies, we can successfully improve the CDR generation from both efficiency and effectiveness perspectives.

Extensive experiments are conducted to verify the value of our method. Specifically, we evaluate on two generation tasks: sequence generation and structure prediction, antigen-binding CDR-H3 design. Compared to existing works, our method achieves the state-of-the-art performance from different evaluation metrics.

Our contributions are summarized as follows: (1) We propose to pre-train a sequence-based antibody language model and successfully incorporate the pre-trained model for antibody sequence-structure co-design. (2) We introduce the one-shot generation way to replace the autoregressive manner so to make the decoding process more efficient. (3) Empirical studies verify the effectiveness and efficiency of our proposed method and the state-of-the-art performance are achieved for antibody design.

2 Related Work

2.1 Antibody Pre-training

Inspired by the great success of protein pre-training [9, 57, 47, 31, 54, 12], a few works attempt to transfer these techniques to the antibody pre-training since the specificity of antibody [51, 50]. Ruffolo et al. [38] first propose an antibody-specific language model AntiBERTy to aid understanding of immune repertoires by training on antibody sequence data. They find the model can cluster antibodies into trajectories resembling affinity maturation and identify key binding residues. Leem et al. [22] introduce AntiBERTa, an antibody-specific bidirectional encoder representation from Transformers, the success is demonstrated by the leading results in antibody binding site prediction and paratope position prediction. Olsen et al. [34] pre-train two antibody models, an Ablang-H model trained on the heavy chain and an Ablang-L model trained on the light chain of antibody. They show the model power on restoring missing residues in antibody sequence data, which also surpasses the general protein pre-training model ESM-1b [37]. Shuai et al. [44] introduce a deep generative language model for generating synthetic libraries by re-designing variable-length spans of antibody sequences. Our work differs from above works that we incorporate the pre-trained sequence model for both sequence and structure predictions of antibody.

2.2 Antibody Design

Antibody design is special to the general protein design. Protein design [32, 14, 48, 5, 19] mainly focuses on the sequence, generation problem that conditioned on a known 3D structures, while antibody design is based on the assumption that both the sequence and the structure of antibody are unknown. Traditional computational antibody design methods mainly utilize the energy function optimization, which use the physics inspired methodology to optimize the sequence and structure of the antibody to reach a minimal energy state [25, 6, 21, 1]. Monte Carlo simulation process is a typical adopted way to search over the energy space. However, these methods seriously suffer from the high cost and low computation efficiency. Recently, the deep learning methods have attracted much attention for antibody design and different generative models have been proposed. Apart from the common works that only try to predict the antibody sequences [3, 43, 39, 2], co-predicting the sentence and the structure of CDRs is more promising but also challenging. How to encode both the 3D structure and sequence information and keep the specific property, e.g., equivariance, are important topics in this area. Jin et al. [16] propose a RefineGNN model to encode the graph features using graph neural network and generate the amino acid sequences and structures in an autoregressive refinement manner. However, RefineGNN only considers the antibody itself without the specificity of antigen-antibody complex. Hence, Jin et al. [17] further introduce HERN to model antibody-antigen docking and design via hierarchical equivariant refinement. HERN employs hierarchical message passing networks to encode both atoms and residues, the prediction is also performed in an autoregressive manner. Differently, our work co-design the CDRs in a one-shot way to avoid the autoregressive generation.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

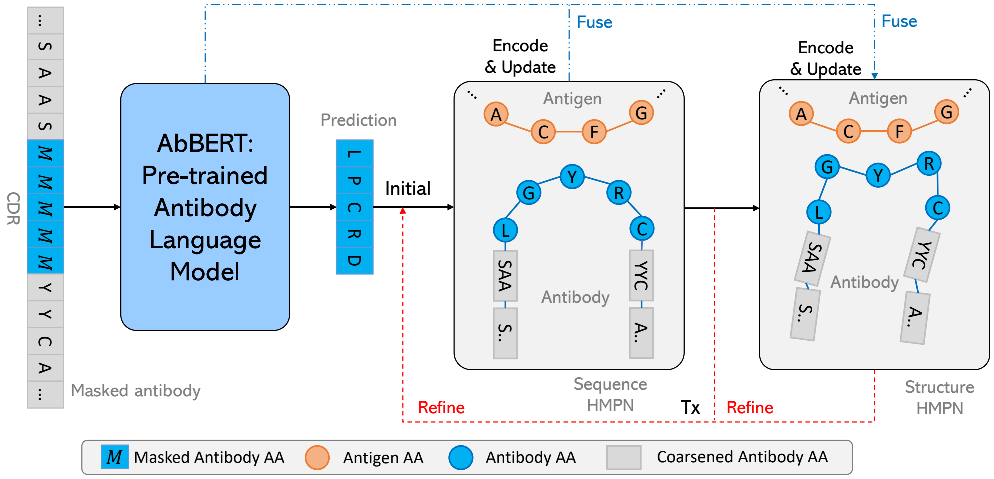


Figure 2: The overall framework of our method. The AbBERT is the pre-trained antibody model. Its 'soft' prediction (for clear illustration, we draw hard prediction here, details in Section 3.3)will be fed into the sequence HMPN $\mathcal{H}_{seq}$, after encoding and generating the updated sequence, structure HMPN $\mathcal{H}_{str}$ encodes the updated graph and then predict the structures. The sequence and structure prediction iteratively refine T times.

3 Methods

In this section, we introduce our method in detail, with the background introduction at first, and then our antibody pre-training, co-designed model and the pre-training leveraging method. Finally we provide the overall algorithm and also some discussions.

3.1 Background and Overview

An antibody consists of two symmetric parts in a Y-shape, which is shown in Figure 1. In each part, there are two chains, a heavy chain and a light chain, each composed of one variable domain (V_H, V_L) and constant domains. The variable domain contains a framework region and three complementarity-determining regions (CDRs). These CDRs are the most important regions that can determine the binding affinity and the binding sites are located in CDRs. The CDRs on the heavy chain are denoted as CDR-H1/H2/H3, each filling with contiguous subsequences. Among them, CDR-H3 plays the most crucial role with the highest variability. Hence, how to precisely predict CDRs, especially CDR-H3, is the main focus of antibody design. Following existing works [16, 43], we define the antibody design as a generation task on the CDR-H1/H2/H3. In our work, we consider the design conditioned on the framework region and also the specific antigen.

We follow previous works to use deep graph generative method [16, 17]. For simplicity, below we introduce the scenario of antigen-conditioned antibody generation [17], which conditions on both antigen and framework region. When no antigen is considered, it is then framework-conditioned. The binding interface of an antigen-antibody complex is composed of an epitope (in antigen) and a paratope (in antibody). The paratope is the sequence of residues in the CDR and the epitope is the sequence of residues closest to the paratope. Given an antibody Ab and an antigen Ag in complex[3], the paratope and epitope residue sequences are denoted as $Ab^p = [Ab_i^p]_{i=1}^n$ and $Ag^e = [Ag_j^e]_{j=1}^m$, the rest framework region of Ab is Ab^f. Since the binding occurs on the epitope of antigen, we ignore other parts in Ag and only consider Ag^e. As for the structures, we take the 3D structures of epitope, paratope and the paratope-epitope interface. Different from Jin et al. [17], we also consider the framework region to enrich the context information and benefit the generation. Hence the whole antibody is considered. The structure of each part is then described as point clouds of atoms.

[3] In the scenario of antibody design, an antibody is usually simply defined as the variable domain in heavy/chain that contains CDR and framework region, since the remaining domains of the chain are constant.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

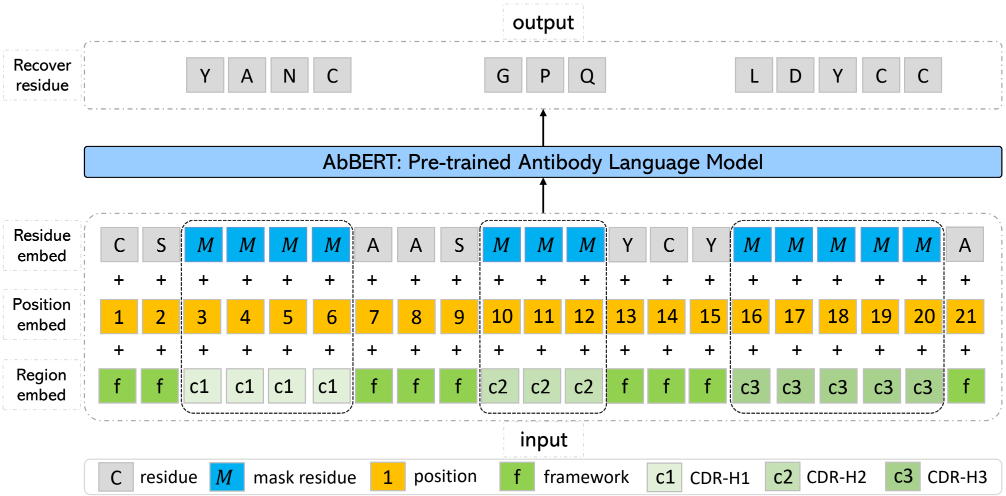


Figure 3: The AbBERT pre-training. The masking is only operated on the CDRs. To indicate the different regions of the variable domain, we set separate region embedding as input for framework (f), CDR-H1 (c1)/H2 (c2)/H3 (c3) regions accordingly.

In an overview, we mainly follow Jin et al. [17] to set the antibody design as a sequence generation and then structure prediction process. The overall framework of our work is shown in Figure 2, which consists of a pre-trained antibody language model AbBERT, a hierarchical message passing network (HMPN) for sequence encoding and generation $\mathcal{H}_{seq}$, and another HMPN for structure prediction $\mathcal{H}_{strc}$. We name our framework as *AbBERT-HMPN*. The encoding/decoding module and the specific model architecture are taken from Jin et al. [17] with necessary modifications from our innovation. The detailed models and the updates will be introduced in Section 3.3.

3.2 AbBERT: Antibody Pre-training

As we discussed, though the antibody structure data is quite limited, it is lucky that a large amount of antibody sequences are collected and existed on the web. Previous works on protein presentation learning have demonstrated the great power of pre-training, we hence introduce the sequence pre-training on antibody to learn expressive representations for antibody design, especially for the paratope sequence prediction. We expect our sequence antibody pre-trained model, AbBERT, can also benefit the structure prediction, where the motivation is from the widely acknowledged sense that the sequence of the protein can determine its structure and function.

Our antibody pre-training follows the style of BERT [20], where the pre-training objective is masked language modeling. However, different from the general text sentence pre-training or the general protein pre-training, the antibody sequence is pretty unique as we introduced before. Hence our antibody pre-training distinguishes from them in two views. (1) The antibody contains a heavy and a light chain which are very different. Since the heavy chain is a more critical sequence, we simply take the sequence data from heavy chain for pre-training. Furthermore, along the whole heavy chain, only the variable domain is what we cared most, the pre-training data is specified to the variable domain in heavy chain (V_H). (2) The different regions of the variable domain also motivate us to pay different attentions to these regions. As the CDRs are the ones that can determine the binding affinity for the antibody, and the framework region serves as the contextual anchor, we set up the masked positions to be the CDRs of the variable domain, and the goal is to recover the amino acids of these masked ones in CDRs.

Specifically, given the antibody sequence $Ab = [Ab_1, ..., Ab_N]$, where N is the number of residues and Ab_i is i-th residue. Instead of random masking, we look at the three CDR continuous subsequences, and we randomly select several residues in these CDR subsequences with some probability p_m and replace these selected residues to be `[MASK]`. Then we will use the expressive Transformer [49] model to encode the masked input antibody sequence with the goal of recovering the masked residues. Denote the original residues for the masked ones as Ab_m, the pre-training objective

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

is to maximize the following log likelihood,

$$L_m(Ab) = \sum\nolimits_{m=1}^{M} \log P(Ab_m|\hat{Ab}), \tag{1}$$

where M is the number of MASK residues and $\hat{Ab}$ is the antibody sequence with the masked residues.

The illustration of our antibody pre-training is presented in Figure 3. There are two different operations compared to the general pre-training [20, 29]. (1) First, the specific functional regions of the variable domain is clearly split, hence besides the residue embedding (similar to the token embedding) and the position embedding, we add an extra region embedding for each residue as the input, e.g., the $f, c1$ in Figure 3. The four region embeddings $f, c1, c2, c3$ are constructed for the framework region, CDR-H1, CDR-H2, and CDR-H3 regions accordingly. (2) Second, since we only take the CDRs as the mask candidates, different from the $p_m = 0.15$ setting in general pre-training, the mask ratio of these reduced candidate space (e.g., CDRs only occupy about 15% of V_H) may require large ratio value for masking. Therefore, we configure p_m to be 0.5 or 1.0 for masking CDRs. The effect of these different p_m is studied in our experiments (see Section 5.4).

After pre-training, our antibody specific language model AbBERT is expected to capture the common knowledge and have expressive representation for antibodies so as to benefit the antibody design in the later stage (see Section 3.4). One notable point is that, thanks to the bidirectional property of the BERT, the recovering of these masked residues are outputted by AbBERT in a one-shot way (simultaneously predict all residues), which ensures not only a good CDR recovering accuracy (from the power of AbBERT) but also an efficient initialization for later antibody design.

3.3 Antibody Sequence-structure Co-design

We treat the co-design as a 3D point cloud completion, or a 3D graph node and structure completion task. The interfaced antibody structures are predicted and the paratope (CDR) sequences are generated. We use hierarchical message passing networks (HMPN) introduced in Jin et al. [17] for encoding and predicting the sequence and structure separately with two HMPN ($\mathcal{H}_{seq}$, $\mathcal{H}_{str}$). As briefly introduced before, our work considers the epitope, paratope and framework for co-design, rather than the antigen and paratope in Jin et al. [17]. Thus, our encoding and decoding processes differs from Jin et al. [17] with necessary modifications.

3.3.1 Encoding

The encoding part in HMPN consists of an atom-level encoding and a residue-level interface encoding to form the hierarchical encoded features for epitope and antibody. The atom-level interface encoding extracts the fine-grained features from the backbone atoms and the side-chains, while the residue-level encoding only captures the backbone C_α atoms. Since we consider the framework region as conditioned context and the framework is usually much longer than CDR, to save the computational cost, we use the coarse-grained encoding proposed by Jin et al. [16]. The framework residues are clustered into residue blocks. The block embedding is the mean of its residue embeddings and the block coordinate is the mean coordinate of its residues,

$$E(b_i) = \sum\nolimits_{Ab_j^f \in b_i} E(Ab_j^f)/b, \quad z(b_{i,c}) = \sum\nolimits_{Ab_j^f} z(Ab_{j,c}^f)/b, \tag{2}$$

where b_i is the block and b is the residue size, E, z stands for the embedding and coordinate for atom, c is the type of atom.

Before introducing the details of the atom-level encoding and the residual-level encoding, let us first describe the initial state of the CDR that we need to predict. Due to the unknown residues and structures, the residues are first initialized by a 'soft' assignment from a learnable feed-forward network in Jin et al. [17] instead of a hard guess. In our work, thanks to the pre-trained AbBERT, the soft assignment of i-th residue is $P(Ab_i^p|\hat{Ab})$ from Eqn.(1), then the node feature is initialized as

$$f(Ab_i^p) = \sum\nolimits_k P(Ab_i^p|\hat{Ab})[k]f(k), \tag{3}$$

where $f(k)$ (introduced below) is the pre-defined amino acid feature for residue type k. For the structures, Jin et al. [17] use a complex distance-based initialization method, instead, we utilize a uniform split between the residue before and after the CDRs of the framework. For example, in

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Figure 3, for CDR-H2, with the coordinates of previous S and later Y denoted as $z(S)$ and $z(Y)$, the initialized coordinates for the inner three residues are $[z(S)-z(Y)]/4$, $[z(S)-z(Y)]*2/4$, $[z(S)-z(Y)]*3/4$ accordingly. With these initialized residues and coordinates, we now introduce the atom-level and residue-level encoding, most of which are following Jin et al. [17].

Atom-level Encoding. The atom-level encoding encodes all the atoms in the graph. For each atom, the node feature is a one-hot encoding of its atom type. The edge feature is the distance encoded by RBF between two atoms $(Ab^p_{i,k}, Ag^e_{j,l})$ in a radial basis, $f(Ab^p_{i,k}, Ag^e_{j,l}) = \texttt{RBF}(||z(Ab^p_{i,k}) - z(Ag^e_{j,l})||)$. Then a MPN is used to learn the feature vectors $h(Ab^p_{i,k}), h(Ag^e_{j,l})$ for atom $Ab^p_{i,k}$ and atom $Ag^e_{j,l}$.

Residue-level Encoding. The residue-level encoding encodes only the C_α atoms in each residue. For each residue, the amino acid feature, e.g., $f(Ab^p_i)$, is first pre-defined by its dihedral angles, polarity, hydropathy and so on (in Appendix A.2). Then the pre-defined feature will concatenate with the sum of the atom feature vectors that outputted from atom-level encoding to form the hierarchical residue node representation, which are defined as:

$$\bar{f}(Ab^p_i) = f(Ab^p_i) \oplus \sum_k h(Ab^p_{i,k}), \bar{f}(Ab^e_j) = f(Ag^e_j) \oplus \sum_l h(Ag^e_{j,l}), \tag{4}$$

where $\oplus$ is concatenation. As for the edge feature, e.g., $f(Ab^p_i, Ab^p_j)$, it is defined as the one that contains the distance, direction, and orientation between two residues, e.g., Ab^p_i, Ab^p_j. With the node feature and edge feature for residues, another MPN is introduced to learn the residue-level hidden representations for epitope and antibody jointly.

3.3.2 Decoding

After hierarchically encoding the residue and atom representations, we can generate the amino acids for CDRs and predict the structures. As shown in Figure 2, we first use $\mathcal{H}_{seq}$ to generate the amino acids, then the generated amino acids will be updated to the $\mathcal{H}_{str}$ for structure prediction.

One big difference between our decoding and Jin et al. [17] is that we adopt an *one-shot* manner instead of the autoregressive way adopted by Jin et al. [17]. The autoregressive generation suffers from two problems. One is the decoding efficiency. The step-by-step decoding is slow, which requires n (length of the CDRs) times decoding, while one-shot decoding only needs one time. Another bad point is the error accumulation problem. During the generation steps, the error from last decoded step will propagate to the current decoding step, which causes the error to be enlarged and resulting wrong predictions, especially for later steps. Though our one-shot manner avoids the error accumulation problem, it can not ensure the accuracy of simultaneously generated residues. Therefore, we add T refinement steps for the one-shot predicted residues and structures to improve the performance. Since T is much smaller than n, the efficiency is guaranteed.

Sequence Generation. The sequence generation step takes the encoded hidden representations. Then the amino acid prediction is performed as a multi-class classification by $\mathcal{H}_{seq}$,

$$P^t(Ab^p_i) = \texttt{softmax}(W_s h^{t-1}(Ab^p_i)), \tag{5}$$

where h^{t-1} is the encoded representation of $t-1$ refinement step. Similar to the initial state of the residue types, we use the 'soft' assignment between the first and the $t-1$ refinement steps, and use this softened version for $\mathcal{H}_{str}$ encoding, only the last step T will output one hard prediction that sampled from the probability for determining the specific amino acid.

Structure Decoding. As discussed in [17], the structure update must maintain the equivariance w.r.t. the rotation and translation of epitope. Hence *force* prediction is utilized instead of the direct coordinate prediction. The force is computed by the residue and atom hidden representations, e.g., $h(Ab^p_i), h(Ab^p_{i,k})$. Specifically, the forces between C_α atoms and other atoms are separately calculated for updating the coordinates. The force between nearest C_α atoms are calculated to update the paratope C_α coordinates. For other atoms, their coordinates are updated according to the force calculated between the atoms in the same residue. Noting that the structure update only performs on the four backbone atoms for paratope residues. The detailed calculation can be found in Appendix A.2 and we give a visual illustration in Figure 4. The updated structures in the $t-1$ refinement step will then feedback to the $\mathcal{H}_{seq}$ for further refinement.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

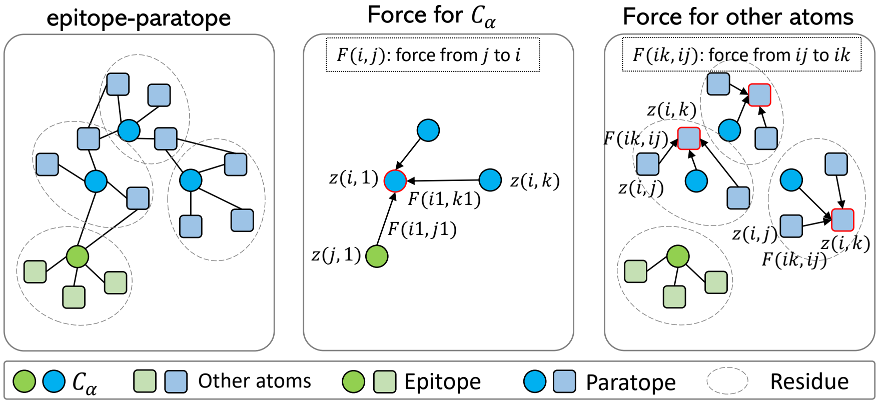


Figure 4: Two kinds of force calculation for the complex (left figure). One is between C_α atoms (middle figure), and the one one is between atoms in the same residue (right figure).

3.4 Incorporating Pre-training

Generally speaking, we introduce two ways to incorporate the pre-trained AbBERT for antibody design. (1) First, we have stated in above subsection that our pre-trained AbBERT provides good initialization for $\mathcal{H}_{seq}$ and $\mathcal{H}_{str}$. The specific initialized way is shown in Eqn.(3), which is to give a soft probability from AbBERT. This soft initialization is supposed to be better than random guess. (2) Second, for both $\mathcal{H}_{seq}$ and $\mathcal{H}_{str}$, we fuse the hidden representations from AbBERT with the residue representations from the encoded MPNs. That is, for the residue hidden encoding, such as $h^{t-1}(Ab_i^p)$ in Eqn.(5) and the hidden $h^B(\hat{Ab}_i^p)$ before the last prediction layer from AbBERT, we simply concatenate them together and pass a linear transformation,

$$\bar{h}^{t-1}(Ab_i^p) = W_b[h^{t-1}(Ab_i^p) \oplus h^B(\hat{Ab}_i^p)], \tag{6}$$

where $\oplus$ is the concatenation operation, and the fused representation $\bar{h}^{t-1}(Ab_i^p)$ will be used in later sequence generation and structure force prediction.

There are two notable points when incorporating AbBERT. (1) To save the computational cost, in each refinement step t, the representation from AbBERT is fixed as $h^B(\hat{Ab}_i^p)$ without updating, this can stop the multiple gradient backpropagation in the refinement steps and we find this impacts little to the performance. (2) During training, our pre-trained AbBERT is finetuned in a prefix-tuning [26] style. That is, some randomized key and value tokens (prefix) are appended in each attention layer of AbBERT, the parameters of the pre-trained AbBERT are fixed and only the appended prefixes are finetuned. In such a way, the knowledge in AbBERT are transferred to the design model with little training cost. We compare the different finetuning ways in Section 5.3.

3.5 Summary and Algorithm

We first introduce the training losses of our AbBERT-HMPN, which include the sequence generation loss and structure prediction loss [17]. The sequence loss is the commonly-adopted cross entropy loss between the predicted probabilities and the ground-truth sequence. For structure loss, they are defined as the Huber loss of the pairwise distances between all predicted atoms $||z(Ab_{i,k}) - z(Ag_{j,l}^e)||$ and the ground-truth distances.

For a quick summary of our pipeline (shown in Figure 2) and the decoding algorithm (shown in Algorithm 1), we first use AbBERT to encode the masked CDR antibody sequence, and then use the output probability and the initialized structure to initialize the graph state, then the $\mathcal{H}_{seq}$ and $\mathcal{H}_{str}$ will encode the graph, the sequence is first generated by $\mathcal{H}_{seq}$ and then the structure is predicted by $\mathcal{H}_{str}$. The sequence and structure decoding will iterate T steps and finally obtain the refined sequence and structure of CDR.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Algorithm 1 Decoding of AbBERT-HMPN

Finetuned AbBERT $\mathcal{M}_B$, sequence HMPN $\mathcal{H}_{seq}$ and structure HMPN $\mathcal{H}_{str}$

```
Predict the initial sequence from the output of $\mathcal{M}_B$, initialize the structure and build graph $G^0$
for t=0 to T-1 do
    Encode the graph $G^t$ with $\mathcal{H}_{seq}$
    Generate the sequence with $\mathcal{H}_{seq}$
    Update the sequence and encode the sequence updated graph with $\mathcal{H}_{str}$
    Predict the structure with $\mathcal{H}_{str}$
    Update the structure and build the new graph $G^{t+1}$
end for
Output the sequence and structure from $G^T$
```

3.6 Discussion

One can see that our work is based on Jin et al. [17]. Apart from incorporating the AbBERT in the pipeline, our work differs from the following points. (1) The context information. Unlike their work, for the antigen we only consider the epitope instead of the entire antigen, and for antibody we add the framework region besides the CDRs. Since the design is on the antibody, we believe more context information from the antibody is beneficial. (2) The graph initialization. We use our AbBERT output as the residue initialization rather than random guess. Besides, our uniformly split structure initialization also distinguishes from the complex distance-based initialization in Jin et al. [17]. (3) The decoding strategy. Differing from the inefficient autoregressive generation way, our one-shot decoding greatly speed up the inference process with higher accuracy.

4 Experiments

4.1 Overview Settings

To evaluate our approach, we conduct experiments on two generation tasks. Following previous works [16, 17], the first setting is sequence generation and structure prediction (Section 4.3), the second task is antigen-binding CDR-H3 design (Section 4.4).

Baseline Models. Since our work is upon Jin et al. [17], most of our compared baseline models are following theirs, including (1) RosettaAntibodyDesign (RAbD) [1], which is a physics-based method used for sequence generation and energy minimization. (2) A LSTM-based model that only works on sequence generation without structure information [39, 2]. (3) Sequence model. A 3D structure-based MPN encoder and an RNN-based sequence decoder, which is implemented by Jin et al. [17]. (4) AR-GNN. An autoregressive-based graph generation model [15] that predicts the amino acid and then the edge iteratively to form the graph. (5) RefineGNN [16]. A GNN-based model that encodes the residue and structure information and co-designs the sequence and structure with iterative refinement. Noting that this model is antibody only. (6) HERN [17]. The antigen-specific antibody design model with hierarchical equivariant message passing neural networks, which also uses autoregressive decoding and iterative refinement.

Model Settings. Our pre-trained AbBERT is a 12-layer Transformer model, each MPN in the $\mathcal{H}_{seq}$ and $\mathcal{H}_{str}$ consists of 4 message passing layers with hidden dimension 256. The refinement steps T in our method is 5. The training details are in Appendix A.2.

4.2 Pre-training

For pre-training AbBERT, we take the existing antibody sequence data from Observed Antibody Space database (OAS) [33]. We follow [4] to preprocess the OAS dataset, which contains sequence filtration and clustering according to the full-length Fv amino acid sequences. We only take the sequences from heavy chain for pre-training. After preprocessing, there are 118,825,825 sequences remained. We sample 50 millions for pre-training. Then we use a BERT$_{\texttt{base}}$ configuration to train AbBERT. The details about the data, pre-training model, and the performance are in Appendix A.1.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Table 1: Results of the sequence generation and structure prediction task on CDR-H1/H2/H3. PPL, AAR, and RMSD stand for perplexity, amino acid recovery and root mean square deviation. The results of the baselines are reproduced by the released code and models from Jin et al. [16].

Method	**CDR-H1**			**CDR-H2**			**CDR-H3**		
	PPL↓	**RMSD↓**	**AAR↑**	**PPL↓**	**RMSD↓**	**AAR↑**	**PPL↓**	**RMSD↓**	**AAR↑**
LSTM	6.79	-	-	7.21	-	-	9.70	-	-
AR-GNN	6.99	2.87	41.88%	6.84	2.34	41.18%	9.23	3.19	18.93%
RefineGNN	3.90	1.39	34.53%	5.15	1.71	29.68%	7.25	2.62	24.22%
AbBERT-HMPN	**2.15**	**0.91**	**55.56%**	**2.36**	**0.67**	**51.46%**	**6.32**	**2.38**	**31.08%**

4.3 Sequence and Structure Prediction

Data. This task aims to evaluate the general ability of the generative model. The data is from Structural Antibody Database (SAbDab) [8]. To make a fair comparison, we follow Jin et al. [16] that only use the antibody sequences without antibody in this task. We directly take the data that processed by Jin et al. [16], where the data is split to train/valid/test set with $8 : 1 : 1$ ratio according to the CDR cluster. The number of clusters for CDR-H1, CDR-H2 and CDR-H3 are 1266, 1564 and 2325 accordingly.

Results. The evaluation metrics for this task are perplexity (PPL), root mean square deviation (RMSD), and amino acid recovery (AAR) between the predicted antibody and the ground-truth. The PPL and AAR are used to measure the sequence generation accuracy and RMSD is used to measure the structure prediction accuracy. The results of this task is shown in Table 1. From the results, we can see that our sequence generation and structure prediction are significantly better on the three metrics than previous works. Specifically, the PPL and RMSD are reduced in a large margin on three CDRs, on CDR-H1/H2, the RMSD is even smaller than 1.0. The AAR accuracy is hugely improved, for example, on CDR-H1, about 20 points accuracy improvement is achieved, which greatly prove the strong modeling ability of our method.

4.4 Antigen-binding CDR-H3 Design

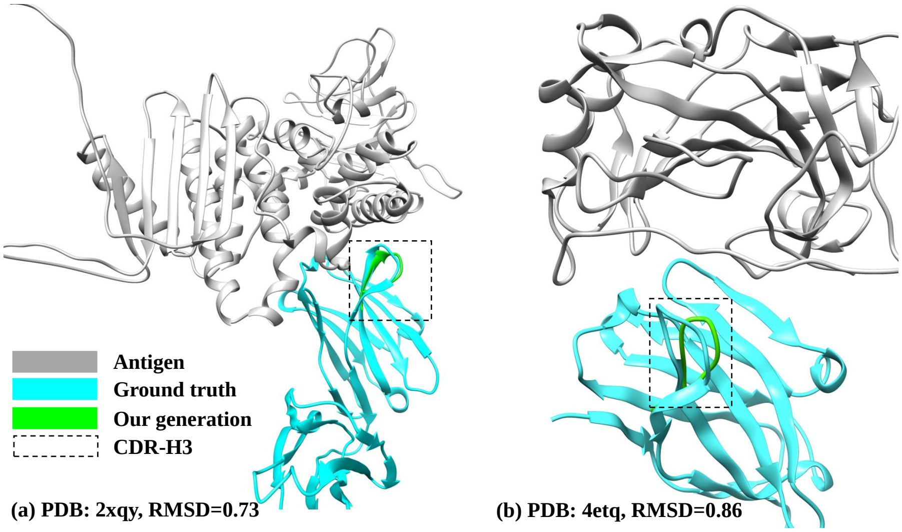
Data. This task is to generate the specific antigen-binding CDR-H3 design, hence the antigen is a conditional input. Specifically, the CDR-H3 is only considered for generation in this task. The testing data is from Adolf-Bryfogle et al. [1], which contains 60 complexes with different antigen types. The training data is also SAbDab dataset, but only the antigen-antibody complexes are utilized. We take the processed data by Jin et al. [17], where the train/valid set contains 2777 and 169 complexes.

Results. Following previous works, we use AAR for evaluation. Besides, we also take RMSD to evaluate the structure prediction accuracy. During inference, we follow Jin et al. [17] to generate 10000 CDR-H3 candidate sequences for each antibody and select the top 100 with the lowest PPL to calculate the average AAR and RMSD. The results are reported in Table 2. We can observe that our AbBERT-HMPN achieves the best results on AAR and RMSD with significant improvement compared to previous works, which demonstrate the strong generation ability and the potential practical value of our method. In detail, our AbBERT obtains 40.35% AAR, which surpasses previous works by more than 6 points. Besides, the RMSD is also largely reduced by about 1.3 point to be 1.62, with almost double performance improvement. We show two cases in Figure 5, where the gray, cyan and green ribbons denote antigen, the ground-truth antibody and our generation respectively. The CDR-H3's are in the dashed box. We can observe that for these two cases, our method can generate CDRs that are close to the ground-truth with RMSD scores 0.73 and 0.86.

Table 2: AAR and RMSD evaluation results for antigen-binding antibody design task. Only CDR-H3 is for generation. The RAbD and sequence model results are taken from Jin et al. [17], while the others are reproduced by ourselves.

Method	**AAR↑**	**RMSD↓**
RAbD	28.6%	-
Sequence model	32.2%	-
AR-GNN	28.07%	2.51
RefineGNN	34.14%	2.99
HERN	33.67%	2.90
AbBERT-HMPN	**40.35%**	**1.62**

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.



(a) PDB: 2xqy, RMSD=0.73

(b) PDB: 4etq, RMSD=0.86

Figure 5: Comparison of the ground truth and generated CDR-H3.

5 Study

In this section, we conduct different study experiments for better understanding our method. Without specific mention, the studies are all performed on antigen-binding CDR-H3 design task with 10 epochs training.

5.1 Ablation Study

We first perform ablation study to investigate the effectiveness of our proposed components. Specifically, we conduct the following experiments: (1) AbBERT-HMPN without the pre-trained AbBERT; (2) AbBERT-HMPN without the framework encoding; (3) AbBERT-HMPN without feature fusing in the $\mathcal{H}_{seq}$, $\mathcal{H}_{seq}$. The results are shown in Table 3. From it, we have following observations: (1) removing each of the above parts will lead to performance drop (AAR becomes smaller and RMSD becomes larger), which proves the positive effect of above components. (2) Among these components, the pre-trained AbBERT contributes most to the final AAR accuracy, and the framework impacts most to the structure. Both rationally demonstrating the design of our method.

Table 3: AAR and RMSD evaluation results for ablation study.

Setting	AAR↑	RMSD↓
AbBERT-HMPN	1.62 %	40.35
−AbBERT	1.84 %	31.82
−framework	2.60 %	37.55
−fusion	1.74 %	36.48

5.2 Effect of Decoding ways

As we mentioned the advantage of one-shot decoding, we study the autoregressive decoding and our one-shot decoding. For easy implementation, we remove the framework region is this study and then do autoregressive or one-shot training and decoding. The results are compared in Table 4. Besides the AAR and RMSD scores, we also provide the memory and time cost during inference to have a better comparison. We can see that our one-shot decoding not only keeps the high efficiency (memory/time cost) but also achieves better performance (AAR/RMSD).

Table 4: Performance and cost comparison between iterative and one-shot decoding.

	AAR↑	RMSD↓	Memory (G)	Time (min)
iterative	32.43 %	3.12	42.4	114
one-shot	37.55 %	2.60	22.1	56

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

5.3 AbBERT Incorporating Ways

In this section, we study the different incorporating ways of the pre-trained AbBERT in our framework, or more specifically, the different tuning methods of the pre-trained AbBERT. We compare the three widely adopted methods, which are (a) fixed without tuning (serves as a feature extractor), (b) all-tuning (all parameters of the pre-trained model are finetuned), and (c) prefix-tuning (only the added prefixes are finetuned while fixing the original model). The results are presented in Table 5. From the numbers, it is obvious that our adopted prefix-tuning performs better than other two ones. This meets our expectation since the structure antibody data is in limited size and the prefix-tuning balances the best between the parameter tuning and knowledge transfer (all-tuning is easy to be overfitting and fixed version is hard to transfer for co-design).

Table 5: Effect of different finetuning ways.

	Fix	**All-tuning**	**Prefix-tuning**
AAR↑/RMSD↓	38.62%/1.83	37.82%/1.86	39.68%/1.82

5.4 Effect of Pre-training

Finally, we also investigate the effect from the pre-training models. That is, we train different settings for the pre-training AbBERT model, and evaluate the final co-design performance to see the effect. Specifically, as discussed before, we choose the mask ratio p_m as a study point and set them to be $[50\%, 80\%, 100\%]$. After finetuning for co-design, the performance are shown in Table 6. We can see the mask ratio indeed impact the co-design performance, and the best setting is 100% masking of the CDRs. This is also reasonable since the CDRs are about 15% of the V_H domain, the pre-training has rich context to learn a good representation for CDR prediction. Hence in the main experiment, we adopt $p_m = 100\%$.

Table 6: Effect of mask ratio p_m of pre-trained AbBERT.

	p_m**=50%**	p_m**=80%**	p_m**=100%**
AAR↑	36.63%	37.15%	39.68%

6 Conclusions

The antibody plays a crucial role in the therapeutic usage in out immune system. The development of the computational antibody design suffers from the limited data issue. In this work, we leverage the large-scale sequence antibody data for pre-training and then adopt the pre-trained model for co-designing the antibody sequences and structures. With some novel designs, our method outperforms strong baselines on different tasks, such as CDR-H3 generation, sequence and structure modeling. In the future, jointly modeling the CDR-H1/H2/H3 and designing more advanced models for sequence-structure co-design are promising.

References

[1] Jared Adolf-Bryfogle, Oleks Kalyuzhniy, Michael Kubitz, Brian D Weitzner, Xiaozhen Hu, Yumiko Adachi, William R Schief, and Roland L Dunbrack Jr. Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS computational biology*, 14(4): e1006112, 2018.

[2] Rahmad Akbar, Philippe A Robert, Cédric R Weber, Michael Widrich, Robert Frank, Milena Pavlović, Lonneke Scheffer, Maria Chernigovskaya, Igor Snapkov, Andrei Slabodkin, et al. In silico proof of principle of machine learning-based antibody design at unconstrained scale. In *Mabs*, volume 14, pp. 2031482. Taylor & Francis, 2022.

[3] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[4] Sharrol Bachas, Goran Rakocevic, David Spencer, Anand V. Sastry, Robel Haile, John M. Sutton, George Kasun, Andrew Stachyra, Jahir M. Gutierrez, Edriss Yassine, Borka Medjo, Vincent Blay, Christa Kohnert, Jennifer T. Stanton, Alexander Brown, Nebojsa Tijanic, Cailen McCloskey, Rebecca Viazzo, Rebecca Consbruck, Hayley Carter, Simon Levine, Shaheed Abdulhaqq, Jacob Shaul, Abigail B. Ventura, Randal S. Olson, Engin Yapici, Joshua Meier, Sean McClain, Matthew Weinstock, Gregory Hannum, Ariel Schwartz, Miles Gander, and Roberto Spreafico. Antibody optimization enabled by artificial intelligence predictions of binding affinity and naturalness. *bioRxiv*, 2022.

[5] Yue Cao, Payel Das, Vijil Chenthamarakshan, Pin-Yu Chen, Igor Melnyk, and Yang Shen. Fold2seq: A joint sequence (1d)-fold (3d) embedding-based generative model for protein design. In *International Conference on Machine Learning*, pp. 1261–1271. PMLR, 2021.

[6] Ratul Chowdhury, Matthew F Allan, and Costas D Maranas. Optmaven-2.0: de novo design of variable antibody regions against targeted antigen epitopes. *Antibodies*, 7(3):23, 2018.

[7] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Christina Floristean, Anant Kharkare, Koushik Roye, Charlotte Rochereau, Gustaf Ahdritz, Joanna Zhang, George M Church, et al. Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, pp. 1–7, 2022.

[8] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

[9] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[10] Darren R Flower, ANTHONY CT North, and Teresa K Attwood. Structure and sequence relationships in the lipocalins and related proteins. *Protein Science*, 2(5):753–761, 1993.

[11] Lorenzo Gentiluomo, Dierk Roessner, Dillen Augustijn, Hristo Svilenov, Alina Kulakova, Sujata Mahapatra, Gerhard Winter, Werner Streicher, Åsmund Rinnan, Günther HJ Peters, et al. Application of interpretable artificial neural networks to early monoclonal antibodies development. *European Journal of Pharmaceutics and Biopharmaceutics*, 141:81–89, 2019.

[12] Liang He, Shizhuo Zhang, Lijun Wu, Huanhuan Xia, Fusong Ju, He Zhang, Siyuan Liu, Yingce Xia, Jianwei Zhu, Pan Deng, et al. Pre-training co-evolutionary protein representation via a pairwise masked language model. *arXiv preprint arXiv:2110.15527*, 2021.

[13] Alissa M Hummer, Brennan Abanades, and Charlotte M Deane. Advances in computational structure-based antibody design. *Current Opinion in Structural Biology*, 74:102379, 2022.

[14] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

[15] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*, pp. 4849–4859. PMLR, 2020.

[16] Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi S Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. In *International Conference on Learning Representations*, 2021.

[17] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Antibody-antigen docking and design via hierarchical structure refinement. In *International Conference on Machine Learning*, pp. 10217–10227. PMLR, 2022.

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

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[19] Mostafa Karimi, Shaowen Zhu, Yue Cao, and Yang Shen. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks. *Journal of chemical information and modeling*, 60(12):5667–5681, 2020.

[20] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[21] Gideon D Lapidoth, Dror Baran, Gabriele M Pszolla, Christoffer Norn, Assaf Alon, Michael D Tyka, and Sarel J Fleishman. Abdesign: A n algorithm for combinatorial backbone design guided by natural conformations and sequences. *Proteins: Structure, Function, and Bioinformatics*, 83 (8):1385–1406, 2015.

[22] Jinwoo Leem, Laura S Mitchell, James HR Farmery, Justin Barton, and Jacob D Galson. Deciphering the language of antibodies using self-supervised learning. *Patterns*, pp. 100513, 2022.

[23] Arthur M Lesk and Cyrus Chothia. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *Journal of molecular biology*, 136(3):225–270, 1980.

[24] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

[25] Tong Li, Robert J Pantazes, and Costas D Maranas. Optmaven–a new framework for the de novo design of antibody variable region models targeting specific antigen epitopes. *PloS one*, 9 (8):e105954, 2014.

[26] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

[27] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pp. 121–137. Springer, 2020.

[28] Edgar Liberis, Petar Veličković, Pietro Sormanni, Michele Vendruscolo, and Pietro Liò. Parapred: antibody paratope prediction using convolutional and recurrent neural networks. *Bioinformatics*, 34(17):2944–2950, 2018.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[30] Igor Melnyk, Payel Das, Vijil Chenthamarakshan, and Aurelie Lozano. Benchmarking deep generative models for diverse antibody sequence design. *arXiv preprint arXiv:2111.06801*, 2021.

[31] Ananthan Nambiar, Maeve Heflin, Simon Liu, Sergei Maslov, Mark Hopkins, and Anna Ritz. Transforming the language of life: transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–8, 2020.

[32] James O'Connell, Zhixiu Li, Jack Hanson, Rhys Heffernan, James Lyons, Kuldip Paliwal, Abdollah Dehzangi, Yuedong Yang, and Yaoqi Zhou. Spin2: Predicting sequence profiles from protein structures using deep neural networks. *Proteins: Structure, Function, and Bioinformatics*, 86(6):629–633, 2018.

[33] Tobias H Olsen, Fergus Boyles, and Charlotte M Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[34] Tobias H Olsen, Iain H Moal, and Charlotte M Deane. Ablang: An antibody language model for completing antibody sequences. *bioRxiv*, 2022.

[35] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[36] Jaime Renart, Jakob Reiser, and George R Stark. Transfer of proteins from gels to diazobenzyloxymethyl-paper and detection with antisera: a method for studying antibody specificity and antigen structure. *Proceedings of the National Academy of Sciences*, 76(7): 3116–3120, 1979.

[37] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[38] Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. Deciphering antibody affinity maturation with language models and weakly supervised learning. *arXiv preprint arXiv:2112.07782*, 2021.

[39] Koichiro Saka, Taro Kakuzaki, Shoichi Metsugi, Daiki Kashiwagi, Kenji Yoshida, Manabu Wada, Hiroyuki Tsunoda, and Reiji Teramoto. Antibody design using lstm based deep generative model from phage display library for affinity maturation. *Scientific reports*, 11(1):1–13, 2021.

[40] Samuel Schmitz, Emily A Schmitz, James E Crowe Jr, and Jens Meiler. The human antibody sequence space and structural design of the v, j regions, and cdrh3 with rosetta. In *Mabs*, volume 14, pp. 2068212. Taylor & Francis, 2022.

[41] Andrew M Scott, Jedd D Wolchok, and Lloyd J Old. Antibody therapy of cancer. *Nature reviews cancer*, 12(4):278–287, 2012.

[42] Sisi Shan, Shitong Luo, Ziqing Yang, Junxian Hong, Yufeng Su, Fan Ding, Lili Fu, Chenyu Li, Peng Chen, Jianzhu Ma, et al. Deep learning guided optimization of human antibody against sars-cov-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11):e2122954119, 2022.

[43] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):1–11, 2021.

[44] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. Generative language modeling for antibody design. *bioRxiv*, 2021.

[45] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

[46] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):2542, 2018.

[47] Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. Udsmprot: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 2020.

[48] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible protein design using deep graph neural networks. *Cell systems*, 11(4):402–411, 2020.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Mai Ha Vu, Rahmad Akbar, Philippe A Robert, Bartlomiej Swiatczak, Victor Greiff, Geir Kjetil Sandve, and Dag Trygve Truslew Haug. Advancing protein language models with linguistics: a roadmap for improved interpretability. *arXiv preprint arXiv:2207.00982*, 2022.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[51] Mai Ha Vu, Philippe A Robert, Rahmad Akbar, Bartlomiej Swiatczak, Geir Kjetil Sandve, Dag Trygve Truslew Haug, and Victor Greiff. Immunolingo: Linguistics-based formalization of the antibody language. *arXiv preprint arXiv:2209.12635*, 2022.

[52] Wei Wang, Satish Singh, David L Zeng, Kevin King, and Sandeep Nema. Antibody structure, instability, and formulation. *Journal of pharmaceutical sciences*, 96(1):1–26, 2007.

[53] Lijun Wu, Xu Tan, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Beyond error propagation: Language branching also affects the accuracy of sequence generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):1868–1879, 2019.

[54] Lijun Wu, Chengcan Yin, Jinhua Zhu, Zhen Wu, Liang He, Yingce Xia, Shufang Xie, Tao Qin, and Tie-Yan Liu. Sproberta: protein embedding learning with local fragment modeling. *Briefings in Bioinformatics*, 2022.

[55] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, 2022.

[56] Wenwu Zhai, Jacob Glanville, Markus Fuhrmann, Li Mei, Irene Ni, Purnima D Sundar, Thomas Van Blarcom, Yasmina Abdiche, Kevin Lindquist, Ralf Strohner, et al. Synthetic antibodies designed on natural sequence landscapes. *Journal of molecular biology*, 412(1):55–71, 2011.

[57] Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Jiazhang Lian, Qiang Zhang, and Huajun Chen. Ontoprotein: Protein pretraining with gene ontology embedding. *arXiv preprint arXiv:2201.11147*, 2022.

[58] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13041–13049, 2020.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

A Appendix

A.1 Pre-training Details

The details of the OAS pre-training data are as follows. OAS currently contains over one billion sequences, from over 80 different studies that cover diverse immune states, organisms and individuals. We firstly exclude sequences which meet criteria as mentioned in [4]. Translated amino acid subsequences of each region are concatenated to make the full-length antibody sequences. Those sequences are clustered using MMseqs2 [45, 46] with the minimum sequence identity set to 0.7 to reduce redundancy. We sample representative sequences of such clusters to make the final pre-training dataset. Tokens in framework regions are masked.

The AbBERT pre-training configuration is similar to $\text{BERT}_{\texttt{base}}$. We use a 12-layer Transformer encoder with hidden dimension 768, feed-forward network size 3072. Adam is the optimizer with initial learning rate $3e-4$, dropout value is 0.1. The training is conducted on 16 V100 GPU cards. As for the performance evaluation, we plot the losses (sequence perplexity) on the training and validation sets along the training process in Figure 6. The training is stable and the loss value shows that the model has learned how to predict the correct token.

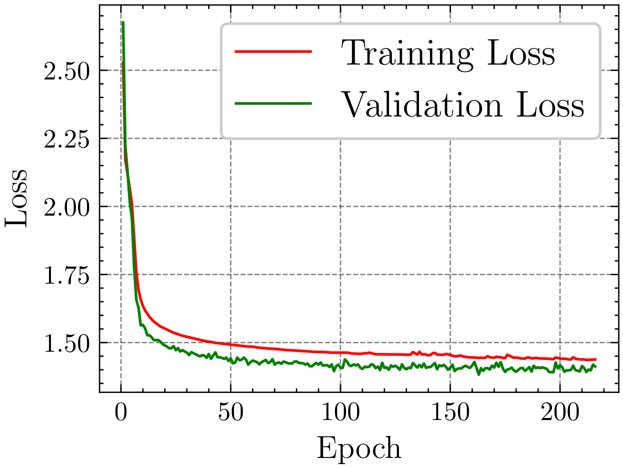


Figure 6: Pre-training loss curves on train and valid sets.

A.2 Model Details

Residue features and residue edge features. We follow Jin et al. [17] to build the initial residue feature and edge feature. Specifically, each amino acid consists of six features: binary polarity, binary hydrogen bond donor, binary acceptor, charge $f_c \in \{-1, 0, 1\}$. Another two hydropathy volume features are expanded into radial basis with interval 0.1 and 10. The total dimension is 112. As for the residue-level edge feature, a local coordinate frame $T_i = [b_i, n_i, b_i \times n_i]$ is defined, then the edge feature $f(a_i, a_j)$ between residue r_i and r_j is

$$f = (E_p(i-j), \texttt{RBF}(||z(i,1) - z(j,1)||), T_i^T \frac{z(j,1) - z(i,1)}{||z(i,1) - z(j,1)||}, q(T_i^T T_j)), \tag{7}$$

where RBF encodes distance, third term encodes the direction information and the last item encodes orientation information.

Message passing network operation (MPN). The message passing network is easy, which is computed as follows:

$$h^{l+1}(r_i) = h^l(r_i) + \sum_{j \in \mathcal{N}_i} \texttt{FFN}(h^l(r_i), h^l(r_j), f(r_i), f(r_i, r_j)), \tag{8}$$

where $h^l(r_i)$ is l-th layer representation for residue r_i, FFN is feed-forward network.

bioRxiv preprint doi: https://doi.org/10.1101/2022.11.14.516404; this version posted November 17, 2022. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Structure force prediction. The structure force between the C_α atoms are calculated by the nearest residues, e.g., epitope-paratope interface,

$$\begin{aligned} F^t(i,j) &= g(h^t(Ab_i^p), h^t(Ab_j^p)) \cdot (z^{t-1}(Ab_{i,1}^p) - z^{t-1}(Ab_{j,1}^p)), \\ F^t(i,k) &= g(h^t(Ab_i^p), h^t(Ag_k^e)) \cdot (z^{t-1}(Ab_{i,1}^p) - z^{t-1}(Ag_{k,1}^e)), \end{aligned} \tag{9}$$

where g is feed-forward network, t is the refinement step. Then the C_α coordinate is updated by the force as

$$z^t(Ab^p(i,1)) = z^{t-1}(Ab^p(i,1)) + \frac{1}{n}\sum_{j \neq i} F^t(i,j) + \frac{1}{m}\sum_k F^t(i,k). \tag{10}$$

For other atoms, the force is calculated among the same residue. For example, for atom $Ab_{i,j}^p$ and $Ab_{i,k}^p$, the force calculation and $Ab_{i,j}^p$ coordinate update is

$$\begin{aligned} F^t(ij,ik) &= g(h^t(Ab_{i,j}^p), h^t(Ab_{i,k}^p)) \cdot (z^{t-1}(Ab_{i,j}^p) - z^{t-1}(Ab_{i,k}^p)), \\ z^t(Ab^p(i,j)) &= z^{t-1}(Ab^p(i,j)) + \frac{1}{n_i}\sum_k F^t(ij,ik). \end{aligned} \tag{11}$$

Training details. For training the co-design model, we use Adam optimizer with seldom hyperparameter search. The learning rate is among $[1e-4, 5e-4]$, epoch ranges in $[10, 20]$. We set number of prefix tokens to be 5, dropout value 0.1.