Reprogramming Pretrained Language Models for Antibody Sequence Infilling

Igor Melnyk [1] **Vijil Chenthamarakshan** [1] **Pin-Yu Chen** [1] **Payel Das** [1] **Amit Dhurandhar** [1] **Inkit Padhi** [1] **Devleena Das** [2]

Abstract

Antibodies comprise the most versatile class of binding molecules, with numerous applications in biomedicine. Computational design of antibodies involves generating novel and diverse sequences, while maintaining structural consistency. Unique to antibodies, designing the complementarity-determining region (CDR), which determines the antigen binding affinity and specificity, creates its own unique challenges. Recent deep learning models have shown impressive results, however the limited number of known antibody sequence/structure pairs frequently leads to degraded performance, particularly lacking diversity in the generated sequences. In our work we address this challenge by leveraging Model Reprogramming (MR), which repurposes pretrained models on a source language to adapt to the tasks that are in a different language and have scarce data – where it may be difficult to train a high-performing model from scratch or effectively fine-tune an existing pre-trained model on the specific task. Specifically, we introduce ReprogBert in which a pretrained English language model is repurposed for protein sequence infilling – thus considers cross-language adaptation using less data. Results on antibody design benchmarks show that our model on low-resourced antibody sequence dataset provides highly diverse CDR sequences, up to more than a two-fold increase of diversity over the baselines, without losing structural integrity and naturalness. The generated sequences also demonstrate enhanced antigen binding specificity and virus neutralization ability. Code is available at `https://github.com/IBM/ReprogBERT`

[1]IBM Research, Yorktown Heights, NY 10598, USA. [2]Georgia Institute of Technology, Atlanta, GA 30332, USA. This work was done during Devleena Das's internship at IBM Research. Correspondence to: Igor Melnyk <igor.melnyk@ibm.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

1. Introduction

Antibodies have emerged as essential therapeutic agents in the treatment of cancer and various other autoimmune, infectious and metabolic diseases. Since 1985, approximately 100 monoclonal antibodies (mAbs) have been designated as drugs by FDA (Jin et al., 2022). Compared to small molecule drugs, the advantage of using antibody proteins as therapeutics is their high specificity resulting in less adverse effects. A key challenge in antibody design is tailoring their binding specificity, which is mainly influenced by the complementarity determining region (CDR). CDR plays a crucial role in antigen recognition and binding processes. It is composed of six hypervariable loops, three formed by each of heavy (H) and light (L) chains. Together, the CDRs shape the antigen binding site of the antibody.

Five of the six loops usually adopt well-characterized canonical conformations. In contrast, the CDR-H3 loop shows substantial variability in sequence and structure, and hence cannot be described by a canonical structure model. When compared to other protein loop structures, the CDR-H3 stands out with its significantly higher structural diversity.

There is a high demand and need for efficient in-silico methods for designing CDRs with improved specificity and other desired properties, to reduce the cost and time associate with wet lab production and testing of antibody candidates. Generative machine learning has emerged as an attractive and viable path for this purpose. For example, for a more general task of protein design, creating new protein sequences that fold to a desired 3D structure and/or exhibit a specific function, many deep generative models have been adapted and expanded (Ingraham et al., 2019; Cao et al., 2021; Karimi et al., 2020; Syrlybaeva & Strauch, 2022; Lee & Kim, 2022; Anand & Achim, 2022). However, compared to other protein design challenges, CDR design (Eguchi et al., 2020; Shin et al., 2021; Adolf-Bryfogle et al., 2018; Fu & Sun, 2022; Kong et al., 2022; Luo et al., 2022), especially CDR-H3 design, comes with additional complexities, such as out-of-distribution generation to accommodate functional novelty. Additionally, in antibody design, sequence similarity may not reflect binding behavior. For example, in HER2 binding antibodies, two very similar sequences (Levenshtein distance < 2) had opposing binding behavior (Mason et al.,

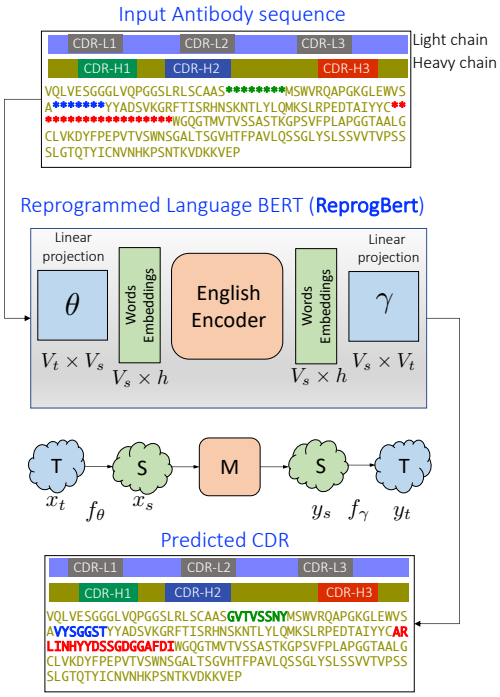

Figure 1. Overview of the proposed Protein Sequence Infilling using Model Reprogramming. Given a heavy chain of an antibody, the goal is to design three Complementarity-Determining Regions (CDR-H1, CDR-H2, CDR-H3), shown in green, blue and red colors, using information from the rest of the protein. The infilling problem is formulated similar to the masked-language modeling task, where the missing amino acids are marked with a token ⟨MASK⟩ and the model generates tokens to infill them. We emphasize that *our system is a sequence-only method*, and while the structure information might be available, our method does not rely on it in the generation process. It makes the model computationally efficient while still achieving high sequence recovery and diversity rates as compared to the current baselines. Reprogrammed language BERT model (*ReprogBert*) is our proposed infilling model, where the English language BERT remains unchanged and frozen (source domain), and we introduce additional amino acid embeddings (target domain) together with the linear matrices ($\theta \in \mathbb{R}^{|V_t|\times|V_s|}$ and $\gamma \in \mathbb{R}^{|V_s|\times|V_t|}$) to project from one domain to another. During CDR infilling training, only the projection matrices and protein embeddings are fine-tuned, the language model remains unmodified. The lower diagram shows the schematic view of the reprogramming: $f_\theta : x_t \to x_s$ is transforming input protein sequence (target domain (T)) into input word sequence (source domain (S)) and $g_\gamma : y_s \to y_t$ reverses the mapping. Thus, for a masked protein sequence x_t we get predicted CDR-infilled antibody $y_t = f_\gamma(M(f_\theta(x_t)))$, where M is the pretrained language model.

2021). Furthermore, it is often desirable to explore new antigen binding modes, when designing antibodies for a target of interest. Such out-of-distribution sample generation remains challenging, particularly in a template-constrained generation scenario.

	Training	Updated Parameters	Results Show
ProtBert	Task-adaptive finetuning	All	Accurate Recovery Low Diversity
EnglishBert	Domain-adaptive finetuning	All	Accurate Recovery Low Diversity
ReprogBert	Language-adaptive reprogramming	$\boldsymbol{\theta}, \boldsymbol{\gamma}$ **only**	Accurate Recovery **High Diversity**

Table 1. Comparison of our proposed methods. ReprogBert stands out as an efficient cross-language approach generating accurate and diverse protein sequences.

Most of the prior works compromise on the sequence and structural diversity in generated CDRs for high amino acid recovery and low root mean square deviation (RMSD) from ground truth structure. Moreover, the sequence-based models typically involve LLM training from scratch on NGS repertoire (Olsen et al., 2022), or GNN training on a small sample of antibody sequence-structure pairs (Jin et al., 2021). The GNN-based models also come with a cost associated with inference, e.g., iterative design of nodes and edges in a graph via autoregressive decoding.

To address these challenges, we propose an alternative sequence-only framework (see Fig. 1 for an overview), that is reprogramming existing out-of-domain English language BERT model (Devlin et al., 2018) toward the protein infilling task, given the rest of the sequence as a template. We term this model *ReprogBert*. Additionally, for our sequence-based infilling task we also consider in-domain specialized protein model *ProtBert* (Elnaggar et al., 2020) as well as the English language BERT (*EnglishBert*), whose out-of-domain language token embeddings are replaced with in-domain amino acid embeddings (see Table 1 and Fig. 5 for details). We compare all proposed infilling methods with physics-based and graph-based generative models on a list of tasks ranging from template-constrained CDR design to CDR sequences with predicted SARS-COV-2 neutralization ability. We show that while ReprogBert enjoys comparable high structural consistency, and lower sequence perplexity, when matched against the baselines, it shows high amino acid CDR recovery while providing additional benefit regarding generating highly diverse CDR sequences. These results suggest the potential of ReprogBert toward on-demand generation of the out-of-distribution sequences in the learning from limited data scenario. The other proposed baseline systems, EnglishBert and ProtBert, achieve high CDR sequence recovery rates with consistent structural integrity, although having a modest sequence diversity performance.

In summary, in this work we: **(i)** propose ReprogBert, a system for protein sequence infilling using model reprogramming for the task of antibody CDR design, **(ii)** show promising performance results as compared to many baselines (including our own proposed ProtBert and EnglishBert

baseline infilling methods) and over multiple benchmarks, where our ReprogBert model upholds structural integrity and sequence recovery, while achieving valuable high diversity of the generated sequences. Moreover, the generated CDR sequences frequently have the lowest perplexity, reflecting their well-formed composition and naturalness. ReprogBert further shows its promise in harder CDR design tasks, can handle multiple CDR infilling at once, and does not need structure template information, and **(iii)** observe high data-efficiency of the reprogrammed model, having only a few training parameters, it can be efficiently trained in the data-scarce domains, such as antibody design, while still leveraging information from large out-of-domain language pretraining.

2. Reprogramming for Protein Sequence Infilling

The field of model reprogramming (MR) has focused on repurposing pretrained machine learning (ML) models for varied ML tasks in different domains. It was firstly proposed in an adversarial setting of stealthy resource alternation in (Elsayed et al., 2018) and later extended to cross-domain resource-efficient transfer learning (Chen, 2022; Neekhara et al., 2022). MR achieves state-of-the-art performance in many tasks, especially in data-limited settings, including reprogramming general images for bio-medical measurements (Tsai et al., 2020), human voice for time-series (Yang et al., 2021), and sentence sentiment for protein property (Vinod et al., 2020). While current MR techniques focus on classification tasks, in our work we seek to extend MR capabilities into generative tasks through reprogramming large pretrained language models for protein sequence infilling. To the best of our knowledge, this work is the first study for such an endeavor.

We also note that the term reprogramming is quite different from fine-tuning. Finetuning aims for task-related (e.g., finetuning a ProtBert model on antibody design task) or domain-specific (e.g., adapting an EnglishBert to the medical domain) adaptation of the pretrained model, while the goal of reprogramming is cross-language adaptation, i.e. English to protein (Ruder, 2021). Further, in finetuning, all of the parameters of the pre-trained models are updated (Howard & Ruder, 2018), or task-specific layers are injected into the model (Houlsby et al., 2019), while in reprogramming the pre-trained model remains frozen. Our approach shares some similarity with prompt-tuning (Lester et al., 2021; Li & Liang, 2021; Hambardzumyan et al., 2021), in which soft prompts composed of continuous embeddings are learned in an end-to-end manner, where the pre-trained model remains frozen. However, so far prompt-tuning is limited to task-specific adaptation of a pre-trained model where the task is in the same language, and the output is also in the source language, whereas reprogramming focuses on cross-language adaptation and outputs in the target language domain.

Given a protein sequence, we propose novel CDR loop design as a form of a template-infilling. The template is provided by the amino acid sequence of the constant region of the antibody, as those are conserved and less likely to change, while the sequences corresponding to CDR can vary and change the structure of the antigen binding interface, resulting into modified antigen affinity and specificity. It should be mentioned though the infilling here is performed to design CDRs of antibodies, the framework can be leveraged to infill any protein sequences.

Figure 1 presents an overview of our proposed framework, *ReprogBert*, the reprogrammed language model for protein sequence infilling. Specifically, we use the pretrained English BERT model (Devlin et al., 2018) (in our experiments, it is the `base-bert-uncased` from HuggingFace) and reprogram it for infilling the CDR part of the antibodies.

The number of tokens in the original language task (i.e., source domain) is denoted by V_s (in our experiments $|V_s| = 30522$ word tokens). The language sentence can then be represented as

$$y_s = \langle w_1, w_2, \ldots, w_n \rangle, \tag{1}$$

where w_i is the word token. The number of tokens in the task of interest (i.e., target domain) is denoted by V_t (in our experiments $|V_t| = 30$ protein tokens: 20 amino acid tokens and 10 auxiliary tokens). The protein sentence can then be represented as

$$x_t = \langle a_1, a_2, \ldots, a_n \rangle, \tag{2}$$

where a_i is an amino acid token. We define two mappings (see bottom plot in Fig. 1)

$$f_\theta : x_t \to x_s, \tag{3}$$

transforming input protein sequence into input word sequence and

$$g_\gamma : y_s \to y_t, \tag{4}$$

reversing the transformation by mapping output word sequence into protein one. Following the approach in (Elsayed et al., 2018; Tsai et al., 2020; Vinod et al., 2020) we constrain these mappings to be linear transformations between the source and target domains. Formally, these mappings are represented as

$$x_s = x_t \theta \tag{5}$$

$$y_t = y_s \gamma, \tag{6}$$

where the linear projection matrices

$$\theta \in \mathbb{R}^{|V_t| \times |V_s|} \tag{7}$$

$$\gamma \in \mathbb{R}^{|V_s| \times |V_t|} \tag{8}$$

are the parameters of the transformations. In particular, focusing on the input projection matrix θ and treating x_t as a one-hot sequence representation of the amino acids of length N, i.e., $x_t \in \mathbb{R}^{N\times|V_t|}$, the sequence representation in the English token domain becomes $x_s \in \mathbb{R}^{N\times|V_s|}$. This representation is then projected onto the embedding matrix of the English Bert model $E \in \mathbb{R}^{|V_s|\times d}$ (d is the latent model dimension):

$$x_s^E = x_s E, \tag{9}$$

and continue the usual processing through the transformer layers and blocks.

During training, all model parameters are fixed and only θ and γ are optimized. Specifically, we update θ and γ with respect to minimizing $\mathcal{L}_{NLL}(y_t, y_t^*)$, the loss between the estimated infilled protein sequence $y_t = f_\gamma(M(f_\theta(x_t)))$, given the CDR-masked anitbody x_t and the ground truth sequence y_t^*.

Note that from (9), the amino acid embeddings $E_{aa} \in \mathbb{R}^{|V_t|\times d}$ can be defined as $E_{aa} = \theta E$. In Section H.1 of appendix we visualize these embeddings and examine their clustering based on biological properties such as electrical charge, hydrophobicity, size, etc. (see Fig. 8 in Appendix). We also note that since x_s in equation (9) is usually a dense vector, where all rows of E are mixed during the projection. To better understand the mapping between amino acids and English tokens, in Section H.2 of appendix we replace the projection with cross-attention mechanism and examine its attention weights (see Fig. 9 in Appendix).

3. Experiments

In this section we present evaluation results of our proposed methods on template constrained CDR design using Structural Antibody Database (SabDab) (Dunbar et al., 2013) and Rosetta Antibody Design (RabD) (Jin et al., 2021), and CoV-AbDab dataset (Raybould et al., 2021) neutralization using the model's generated antibodies.

3.1. Evaluation Metrics

For each input protein sequence in our experiments we generated 100 samples using our infill models. To measure the quality of these samples, we then compute the following evaluation metrics (see Fig. 2 for an illustration). *Amino acid recovery (AAR)* is computed for the specific sequence region of interest (e.g., CDR-H3), measuring the percent of the exact matches between ground truth and the sampled sequences. The range is 0-100, and the higher the AAR, the more accurate the recovery. *Diversity (DIV)*, on the other hand, uses only the sampled proteins to compute the complement of the average recovery of all pairwise comparisons in the set. Here the range is 0-100 and the higher the number,

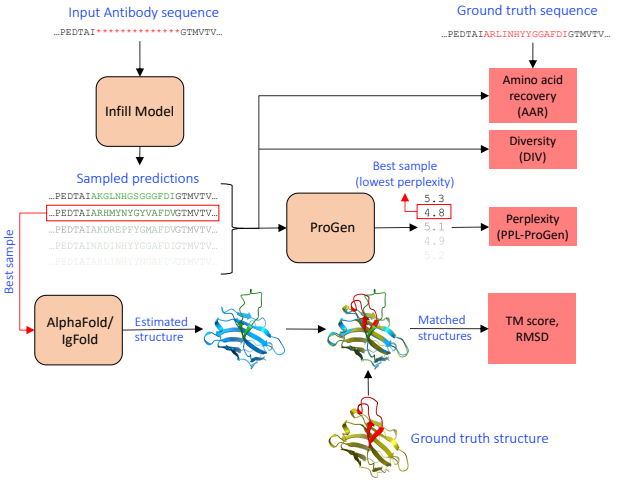

Figure 2. Evaluation process and the computed metrics. For each masked antibody input sequence we generate 100 predicted samples. Metrics are Amino acid recovery (AAR); Diversity (DIV) – dissimilarity of each sample to all the others; Perplexity (PPL-ProGen) computed using off-the-shelf ProGen model (Nijkamp et al., 2022), reflecting "naturalness" of the designed sequences. The sample with the minimum perplexity (red box with an arrow) is then used for 3D structure prediction using AlphaFold (Jumper et al., 2021) or IgFold (Ruffolo et al., 2022) models and compared with ground truth to compute template modeling (TM) score (Zhang & Skolnick, 2004) and the root mean squared deviation (RMSD) from the input structure.

CDR	Train	Validation	Test	Average CDR length	Average CDR diversity
CDR-H1	4050	359	326	8.1	60.8
CDR-H2	3876	483	376	7.9	68.2
CDR-H3	3896	403	437	14.5	76.9

Table 2. Statistics of the Structural Antibody Database (SabDab) for the training, validation and test splits across the three CDRs. We also show the average number of amino acids per CDR and average CDR diversity (length-normalized) across proteins. As can be seen CDR-H3 is the longest and most diverse and therefore represents the most challenging prediction task.

the more dissimilar are samples among themselves. While in general it holds true that the recovery and diversity are inversely correlated, i.e., higher recovery rate leads to lower diversity, and vice versa, CDR design calls for generative models that achieve at least above 30% recovery (Weitzner et al., 2015), while at the same time are able to maintain high sequence diversity.

3.2. Baseline Models

For *perplexity* (the model's predicted probabilities for every residue in a given sequence) we use off-the-shelf autoregressive Transformer protein model ProGen (Nijkamp et al., 2022) to compute *PPL-ProGen* as the average of 100 samples (masking only the region of interest). Specifically, we

used ProGen2-small (151M parameters), which has been pretrained on the mixture of Uniref90 (Suzek et al., 2015) and BFD30 (Steinegger & Söding, 2018) datasets. For perplexity, the lower values mean the better performance, indicating stronger "naturalness" of generated CDRs. The sampled protein sequence with the minimum perplexity is then used for 3D structure prediction using protein folding model (e.g., AlphaFold (Jumper et al., 2021) or IgFold (Ruffolo et al., 2022)). The full predicted and ground truth structures are then compared to compute *template modeling (TM)* score (Zhang & Skolnick, 2004) (range 0-100, higher the better) and the *root mean squared deviation (RMSD)* (lower the better), focusing only on the CDR part. The suffix AF correpsonds to AlphaFold, while IF means IgFold.

We included the following baseline methods to compare against our BERT-based infilling models. *LSTM* from (Saka et al., 2021) and (Akbar et al., 2022), which, similar to ours, is a sequence-only model, however of smaller capacity, having a single attention layer between the input and output layers. *AR-GNN* - autoregressive graph neural network (Jin et al., 2021), which is a sequence and structure-based model, at each step first it predicts the amino acid, followed by the edge generation between the current and all the past residues. *RefineGNN* (Jin et al., 2021) is a model that designs protein sequence and 3D structure of CDR together as graphs. At each step the method predicts residues autoregressively and simultaneously refines the predicted global structure, which in turn helps in subsequent residue prediction. To improve computational efficiency, they employ coarse-grained modeling by clustering every predefined number of context residues in a block, thus reducing the size of the computational graph. *AbLang* (Tobias H. Olsen & Deane, 2022), a language model trained on the antibody sequences in the Observed Antibody Space (OAS) (Kovaltsuk et al., 2018), and which was designed to restore missing residues in antibody sequences.

Additionally, to better evaluate the proposed ReprogBert model, we propose our own baseline approaches for the sequence-based infilling task. The first is in-domain protein model *ProtBert* (Elnaggar et al., 2020), which focuses on task adaptation. It is a specialized protein model that has been pretrained on millions of protein sequences and therefore is well suited for antibody CDR infilling task. The second is the English language BERT (*EnglishBert*) model, whose out-of-domain language token embeddings are replaced with in-domain amino acid embeddings (see Fig. 5 in Appendix for details), thus the goal here is the domain adaptation.

We emphasize that among the three proposed infilling approaches, during training all the parameters of the ReprogBert remain frozen except the two linear projection matrices θ and γ, which are learned and optimized. On the other hand, all the parameters of ProtBert and EnglishBert, as in typical finetuning, are still updated and optimized for the CDR infilling task.

3.3. Structural Antibody Database (SabDab)

SabDab (Dunbar et al., 2013) is a dataset containing antibody sequences and the corresponding 3D structure information, annotated with several properties like gene details, heavy and light chain pairings, CDR location, etc. For this experiment, we used the dataset curated by (Jin et al., 2021) and the statistics are shown in Table 2. The evaluation results are shown in Tables 3, 4, and 5. We note that the values for PPL and RMSD metrics for LSTM, AR-GNN and Refine-GNN are from the published results (Jin et al., 2021). Comparing across the three experiments, we can see that CDR-H1, CDR-H2 and CDR-H3 estimations are progressively harder problems, which is reflected in the drop of AAR across all the methods. Among the proposed infill methods, Except LSTM and AR-GNN, all models achieve over 30% AAR, implying consistency with the ground-truth sequence. ReprogBert has a good recovery accuracy and at the same time generates very diverse and lowest perplexity CDR sequences, while finetuned Bert models generate less diverse and higher perplexity sequence consistent with earlier-reported performance gap of finetuning. We emphasize that the performance of the Bert-based models is without the access to the available 3D structure information. RefineGNN, on the other hand, using both sequence and structure constraints, overall preforms competitively, generating CDR sequences that are accurate and diverse. Nevertheless, the advantage of ReprogBert is more prominent for longer CDR-H3, which is the hardest design task of all three, where ReprogBert evidently outperforms RefineGNN in term of perplexity, AAR, and diversity, while maintaining structural integrity. Also observe that AbLang has a consistently lower recovery rate, and even dropping below 30% threshold, particularly for longer CDR-H3. Such performance might be partially due to a mismatch in the training data of AbLang. Finally, in Fig. 6 we show the results of all three CDRs infilling at once. The BERT-based models are not architecturally limited to a single CDR generation, in contrast to Refine-GNN, therefore can infill multiple regions at once with similar high recovery, structural consistency, and diversity scores.

Since our BERT-based infill models do not estimate protein structure, we use AlphaFold (Jumper et al., 2021) and IgFold (Ruffolo et al., 2022) to estimate 3D structure from the generated sequence and compute TM and RMSD scores with respect to groundtruth native structure. We can see from the Tables 3, 4, 5, and 6 that all the methods have similar structural consistency results (TM and RMSD-AF). However, these values are consistently higher when compared to RMSD for "natively" predicted structure (AR-GNN and

	SabDab CDR-H1								
	PPL	PPL-ProGen	RMSD	RMSD-AF	RMSD-IF	TM-AF	TM-IF	AAR	DIV
LSTM	6.79	–	–	–	–	–	–	–	–
AR-GNN	6.47	–	2.97	–	–	–	–	–	–
Refine-GNN	6.09	3.5	1.18	4.42	1.78	84.0	93.6	61.2	47.3
AbLang	–	–	–	–	–	–	–	47.7	42.8
ProtBert	–	3.5	–	4.16	1.68	84.4	93.8	64.7	4.6
EnglishBert	–	3.7	–	4.22	1.67	84.1	93.8	63.6	5.8
ReprogBert	–	3.3	–	4.31	1.73	84.0	93.7	56.0	29.1

Table 3. Evaluation results on the SabDab dataset for CDR-H1 in the heavy chain. Dark grey cell denote best results, while light grey are the second best. ReprogBert generates sequences with lowest perplexity, second best diversity and high AAR with structural consistency.

	SabDab CDR-H2								
	PPL	PPL-ProGen	RMSD	RMSD-AF	RMSD-IF	TM-AF	TM-IF	AAR	DIV
LSTM	7.21	–	–	–	–	–	–	–	–
AR-GNN	6.86	–	2.27	–	–	–	–	–	–
Refine-GNN	6.58	3.4	0.87	3.05	1.40	85.7	93.9	48.9	38.7
AbLang	–	–	–	–	–	–	–	46.7	44.9
ProtBert	–	3.6	–	3.10	1.32	85.6	93.9	59.5	5.5
EnglishBert	–	4.0	–	3.07	1.32	85.6	93.9	59.1	7.7
ReprogBert	–	3.9	–	3.02	1.40	85.8	93.8	53.0	37.9

Table 4. Evaluation results on the SabDab dataset for CDR-H2 in the heavy chain. As compared to Table 3, all of our proposed infill methods now outperform RefineGNN in terms of AAR metric, while reprogBert also provides second best diversity.

	SabDab CDR-H3								
	PPL	PPL-ProGen	RMSD	RMSD-AF	RMSD-IF	TM-AF	TM-IF	AAR	DIV
LSTM	9.20	–	–	–	–	–	–	–	–
AR-GNN	9.44	–	3.63	–	–	–	–	–	–
Refine-GNN	8.38	7.2	2.50	5.62	3.43	85.0	94.0	28.2	25.7
AbLang	–	–	–	–	–	–	–	22.0	71.3
ProtBert	–	6.8	–	5.40	3.39	85.2	94.0	41.5	14.5
EnglishBert	–	5.9	–	5.53	3.26	84.9	94.0	35.6	59.8
ReprogBert	–	5.4	–	5.54	3.44	85.1	94.0	32.6	67.4

Table 5. Evaluation results on the SabDab dataset for CDR-H3. As compared to CDR-H1 (Fig. 3) and CDR-H2 (Fig. 4), longer CDR-H3 design is more challenging, which shows a drop in AAR across all the methods. ReprogBert clearly outperforms RefineGNN on this hard task, as evident from lower PPL, better AAR, and better diversity.

Refine-GNN), which is likely due to the estimation errors introduced by the AlphaFold or IgFold algorithm. Since RefineGNN focuses on recovering both groundtruth sequence and structure, it does so by sacrificing exploration of the broader sequence space accessible to a given structure (Tian & Best, 2017), which is not the case for ReprogBert.

To further qualitatively illustrate the effect of recovery and diversity on the sampled sequences, we show in Fig. 3 AlphaFold-generated 3D structures of the protein sequences generated by the ReprogBert model. High structural diversity of the CDR-H3 is clearly visible by the coverage of the CDR-H3 ensemble (ground-truth shown using opaque while generated shown as transparent) . Fig. 4 presents a visualization of sequence similarity (in green)/diversity (in white to blue) across models. For example, for ProtBert the third column has a residue D in all the rows (high frequency), thus having the darkest shade, while for ReprogBert the last column has only two generated Y's (low frequency), thus colored in the light shade of blue. Therefore, the method with the high recovery and high diversity rates will have many green and light blue cells. Comparing with Table 5, we indeed see that ReprogBert has highest diversity represented by the largest number of light blue cells, at the same time ProtBert has most green cells (highest AAR), but also many dark blue cells (low diversity). It can also be seen that RefineGNN has lower diversity and lower recovery, as compared to ReprogBert. Further, the 2D kernel density plot as a function of isoelectric point (pH when net charge is 0) and length of CDR-H3 shown in Figure 13 in Appendix implies ReprogBert maintains highest physicochemical similarity to the natural CDRs.

	SabDab CDR-H1,2,3						
	PPL-ProGen	RMSD-AF	RMSD-IF	TM-AF	TM-IF	AAR	DIV
AbLang	–	–	–	–	–	40.1	54.2
ProtBert	4.9	4.8	2.62	85.0	94.3	57.6	8.2
EnglishBert	5.2	4.83	2.62	85.0	94.2	56.3	8.3
ReprogBert	3.9	4.95	2.73	84.8	94.0	42.4	57.4

Table 6. Evaluation results on challenging task of generating the three heavy chain CDR loops at once using the SabDab dataset. The reprogrammed model showed the lowest perplexity, good structural consistency, and the highest sequence variability .

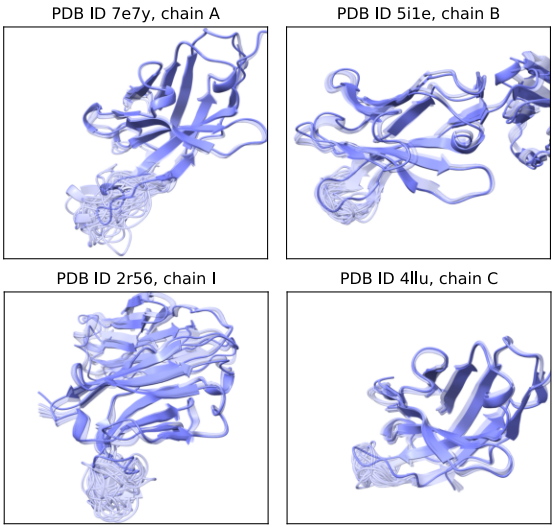

Figure 3. AlphaFold-estimated 3D structures of the proteins generated by the ReprogBert model on SabDab dataset. Each plot shows 30 generated CDR samples for a specific PDB ID. The ground truth and the generated CDR are shown on the bottom part of each figure using solid and faded colors, respectively. As can be seen, CDR-H3 part shows high structural diversity, confirming the same findings as in Table 5, i.e., that ReprogBert achieves high recovery rate while maintaining the highest sequence diversity.

Dataset	CDR	Train	Validation	Test	Average CDR length
RabD	CDR-H3	8646	98	58	14.5
CoV-AbDab	CDR-H3	2282	291	291	15.7

Table 7. Statistics of Rosetta Antibody Design (RabD) and Coronavirus Antibody Database (CoV-AbDab) datasets for CDR-H3.

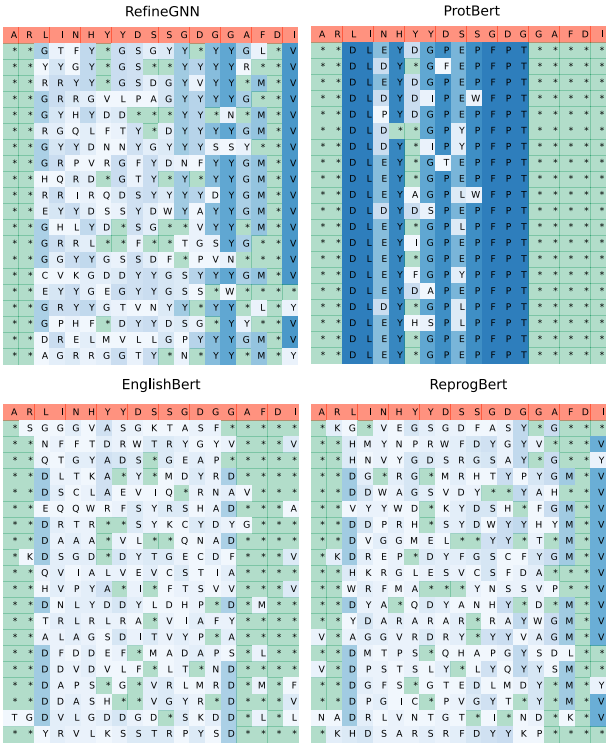

Figure 4. Visualization of the sequence recovery and diversity metrics for generated CDR-H3 (PDB ID 7e7y) across different models. The top row (in red) shows the ground truth CDR-H3 , while the following 20 rows correspond to the generated CDR-H3s. The green cell with the star symbol represents the same amino acid as in the ground truth, while the white/blue cell shows new and different generated residues. The darker shade of the blue cell represents the frequency of the amino acid in that column.

3.4. Antigen-Specific Antibody Design

The goal here is to design a CDR that binds a given antigen, given the antibody sequence template. For this experiment, we used the dataset curated by (Jin et al., 2021), statistics of which is shown in Table 7. In particular it consists of all the SabDab 7 for training, excluding sequences in the same cluster as test antibodies, which were proposed by (Adolf-Bryfogle et al., 2018). In addition to the earlier defined baselines, for this experiment, similar to (Jin et al., 2021), we compared against a physics-based baseline, RabD (Adolf-Bryfogle et al., 2018), which first grafts a CDR from an internal database into the groundtruth antibody structure, followed by iterations of amino acid substitutions and energy minimization. The results are shown in Table 8. The values for PPL, RMSD, and AAR metrics for RabD, LSTM, AR-GNN and Refine-GNN baselines are from (Jin et al., 2021). ReprogBert shows the best diversity rate with accurate sequence recovery and structural consistency.

	RabD CDR-H3								
	PPL	PPL-ProGen	RMSD	RMSD-AF	RMSD-IF	TM-AF	TM-IF	AAR	DIV
RabD	9.20	–	–	–	–	–	–	28.53	–
LSTM	9.20	–	–	–	–	–	–	22.53	–
AR-GNN	9.44	–	3.63	–	–	–	–	23.86	–
AbLang	–	–	–	–	–	–	–	21.3	70.9
Refine-GNN	8.38	4.7	2.50	5.06	2.52	82.9	96.0	35.4	31.1
ProtBert	–	7.7	–	5.42	2.35	82.3	96.2	53.1	11.6
EnglishBert	–	7.8	–	5.34	2.19	82.4	96.3	54.9	10.1
ReprogBert	–	5.1	–	4.72	2.47	83.0	96.1	36.3	62.1

Table 8. Evaluation results on the RabD dataset for CDR-H3. The ReprogBert achieves the best diversity rate with accurate sequence recovery and structural consistency.

	Training on CoV-AbDab			Training on CoV-AbDab + SabDab		
	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV
ProtBert	6.0	50.7	13.6	7.8	49.6	10.7
EnglishBert	6.3	49.3	9.5	8.2	49.2	11.0
ReprogBert	5.7	39.3	60.2	4.9	37.3	64.1

Table 9. Evaluation results on the CoV-AbDab dataset for generated CDR-H3. Since no ground truth structure is available for this dataset, the structure consistency metrics are not computed.

3.5. Coronavirus Antibody Database (CoV-AbDab)

We also show generality of our reprogramming approach on CoV-AbDab (Raybould et al., 2021), a public database documenting all published and patented antibodies and nanobodies able to bind to coronaviruses, including SARS-CoV2 and SARS-CoV1. We used the dataset curated by (Jin et al., 2021) (see Table 7). The evaluation results are shown in Table 9, where only the sequence-based metrics are presented since the ground truth structure information is unavailable for this task. The results are presented for the case of training only on CoV-AbDab and the case of training on both CoV-AbDab and SabDab datasets, showing overall similar trend, i.e. ReprotBert achieving the highest diversity while maintaining good sequence recovery and low perplexity.

The second step of our evaluation is to measure the ability of the generated antibodies to neutralize SARS-CoV2 virus, for which we follow the setup of (Jin et al., 2021). Specifically, we employ the neutralization classifier, composed of SRU encoder (Lei, 2021), pooling and feed-forward network, as provided in (Jin, 2022), together with the iterative target augmentation (ITA) framework (Yang et al., 2020). The goal is to additionally fine-tune the infilling models to generate CDRs resulting into better neutralizing antibodies, as measured by the classifier. Table 10 presents the results. Note that the performance values for the neutralization classifier, LSTM, AR-GNN and Refine-GNN are from (Jin et al., 2021), for which they pretrained these models on SabDab dataset followed by the training on CoV-AbDab. As can be seen from the table, under both training scenarios, our ReprogBert infilling method gets the largest improvement over the original neutralization classifier, achieving 75.6 % and 76.7 % neutralization scores, respectively.

	Neutralization Score	
Model	CoV-AbDab	CoV-AbDab + SabDab
Original	–	69.3
LSTM	–	72.0
AR-GNN	–	70.4
Refine-GNN	–	75.2
ProtBert	72.7	74.7
EnglishBert	70.5	71.0
ReprogBert	75.6	76.7

Table 10. Neutralization of SARS-CoV-2 virus as predicted by the pre-trained SARS-CoV-1 / SARS-CoV-2 classifier. The neutralization score is defined as the predicted probability of a given antibody to neutralize the SARS-CoV-2 virus, as measured by the neutralization classifier.

4. Limitations of our work

In this section, we discuss the challenges faced by our proposed model in handling larger protein infilling tasks, its dependence on pre-trained language models, and the potential limitations stemming from a restricted training dataset which may impact the experimental validation of generated antibody sequences in wet labs.

- Limited performance on larger protein infilling tasks: While the proposed model shows promising results on smaller CDR regions, it may not perform well on larger protein infilling tasks, when the context becomes too small, and where more complex structural depen-

dencies need to be considered. However, the better performance on longer CDR-H3 of ReprogBert, compared to baselines, is noteworthy and promising for extending it to loop design task in general. One way to handle that would be to force attention close to the region to be infilled, which will be future work.

- Dependence on pre-trained language models: The proposed model relies on existing pre-trained language models, which may not capture all the domain-specific features required for protein infilling tasks. Reprogramming provides a framework to learn mapping between amino acids and English vocabularies, given that the protein sequences capture structure and functional information (Rives et al., 2021) and, protein sequences share linguistic features such as Zipfian distribution and existence of grammar (Yu et al., 2019).
- Since the model was trained on a limited set of available antibody data, it may not capture all the complex interactions and structural constraints required for functional antibodies. The generated antibody sequences therefore may fail experimental validation of wet lab.

5. Conclusion

In this work we introduced ReprogBert, a reprogramming framework leveraging pretrained English language models for protein sequence infilling. Specifically, we formulated variable CDR loop design as a template-infilling, where the template is provided by the constant region of the antibody. Results show promising performance, when compared to existing sequence and graph-based deep generative baselines, as well as straightforward task-specific and domain-specific finetuned pre-trained language models. Over multiple benchmarks, our ReprogBert model upholds structural integrity, sequence recovery, and naturalness, while achieving high novelty and diversity of the generated sequences. The improvement is more obvious for the longer CDR-H3. ReprogBert can also handle multiple CDR infilling at once with losing performance. The generated sequences demonstrate enhanced antigen binding specificity and virus neutralization ability in silico. Analysis of the amino acid embeddings learned by ReprogBert from English token embeddings reveals efficient mapping between the two domains as well as naturally occurring clustering of amino acids with meaningful biological properties. Finally, it is worth emphasizing the high data-efficiency of the reprogrammed model, which results from having only a few training parameters (consisting of two linear projection matrices) that can be efficiently trained in the data-scarce domains, such as antibody design, while still leveraging information from large out-of-domain language pretraining. This advantage allows the reprogrammed English language model to efficiently and effectively adapt to the antibody sequences, and perform competitively or better with respect to other supervised or finetuned baselines that either learn from both sequences and structures, or requires more expensive finetuning, or show performance degradation on chanllening design tasks.

References

Adolf-Bryfogle, J., Kalyuzhniy, O., Kubitz, M., Weitzner, B. D., Hu, X., Adachi, Y., Schief, W. R., and Dunbrack Jr, R. L. RosettaAntibodyDesign (RAbD): A general framework for computational antibody design. *PLoS computational biology*, 14(4):e1006112, 2018.

Akbar, R., Robert, P. A., Weber, C. R., Widrich, M., Frank, R., Pavlović, M., Scheffer, L., Chernigovskaya, M., Snapkov, I., Slabodkin, A., et al. In silico proof of principle of machine learning-based antibody design at unconstrained scale. In *Mabs*, volume 14, pp. 2031482, 2022.

Amimeur, T., Shaver, J. M., Ketchem, R. R., Taylor, J. A., Clark, R. H., Smith, J., Van Citters, D., Siska, C. C., Smidt, P., Sprague, M., et al. Designing feature-controlled humanoid antibody discovery libraries using generative adversarial networks. *BioRxiv*, 2020.

Anand, N. and Achim, T. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Anand, N. and Huang, P. Generative modeling for protein structures. *Advances in neural information processing systems*, 31, 2018.

Cao, Y., Das, P., Chenthamarakshan, V., Chen, P.-Y., Melnyk, I., and Shen, Y. Fold2seq: A joint sequence (1d)-fold (3d) embedding-based generative model for protein design. In *International Conference on Machine Learning*, pp. 1261–1271, 2021.

Chen, P.-Y. Model reprogramming: Resource-efficient cross-domain machine learning. *arXiv preprint arXiv:2202.10629*, 2022.

Das, P., Sercu, T., Wadhawan, K., Padhi, I., Gehrmann, S., Cipcigan, F., Chenthamarakshan, V., Strobelt, H., Dos Santos, C., Chen, P.-Y., et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering*, 5(6):613–623, 2021.

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

Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. SAbDab: the

structural antibody database. *Nucleic Acids Research*, 42 (D1), 11 2013.

Eguchi, R. R., Anand, N., Choe, C. A., and Huang, P.-S. IG-VAE: Generative modeling of immunoglobulin proteins by direct 3d coordinate generation. *bioRxiv*, 2020.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., BHOWMIK, D., and Rost, B. ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020.

Elsayed, G. F., Goodfellow, I., and Sohl-Dickstein, J. Adversarial reprogramming of neural networks. *arXiv preprint arXiv:1806.11146*, 2018.

Fu, T. and Sun, J. Antibody complementarity determining regions (CDRs) design using constrained energy model. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 389–399, 2022.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Greener, J. G., Moffat, L., and Jones, D. T. Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, 8(1):1–12, 2018.

Hambardzumyan, K., Khachatrian, H., and May, J. WARP: Word-level Adversarial ReProgramming. 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)*, August 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019.

Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2018.

Huang, P.-S., Ban, Y.-E. A., Richter, F., Andre, I., Vernon, R., Schief, W. R., and Baker, D. RosettaRemodel: a generalized framework for flexible backbone protein design. *PloS one*, 6(8):e24109, 2011.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Jin, S., Sun, Y., Liang, X., Gu, X., Ning, J., Xu, Y., Chen, S., and Pan, L. Emerging new therapeutic antibody derivatives for cancer treatment. *Signal Transduction and Targeted Therapy*, 7(1):1–28, 2022.

Jin, W. GitHub repository for Iterative refinement graph neural network for antibody sequence-structure co-design (RefineGNN), 2022. URL `https://github.com/wengong-jin/RefineGNN`.

Jin, W., Wohlwend, J., Barzilay, R., and Jaakkola, T. Iterative refinement graph neural network for antibody sequence-structure co-design. *arXiv preprint arXiv:2110.04624*, 2021.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Karimi, M., Zhu, S., Cao, Y., and Shen, Y. 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.

Khan, A., Cowen-Rivers, A. I., Deik, D.-G.-X., Grosnit, A., Dreczkowski, K., Robert, P. A., Greiff, V., Tutunov, R., Bou-Ammar, D., Wang, J., et al. AntBO: Towards real-world automated antibody design with combinatorial bayesian optimisation. *arXiv preprint arXiv:2201.12570*, 2022.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kong, X., Huang, W., and Liu, Y. Conditional antibody design as 3D equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

Kovaltsuk, A., Leem, J., Kelm, S., Snowden, J., Deane, C. M., and Krawczyk, K. Observed Antibody Space: A Resource for Data Mining Next-Generation Sequencing of Antibody Repertoires. *The Journal of Immunology*, 201(8):2502–2509, 2018.

Leaver-Fay, A., Tyka, M., Lewis, S., Lange, O., Thompson, J., Jacak, R., Kaufman, K., Renfrew, P., Smith, C.,

Sheffler, W., Davis, I., Cooper, S., Treuille, A., Mandell, D., Richter, F., Ban, Y., Fleishman, S., Corn, J., Kim, D., Lyskov, S., Berrondo, M., Mentzer, S., Popović, Z., Havranek, J., Karanicolas, J., Das, R., Meiler, J., Kortemme, T., Gray, J., Kuhlman, B., Baker, D., and Bradley, P. *Rosetta3: An object-oriented software suite for the simulation and design of macromolecules*, pp. 545–574. 2011.

Lee, J. S. and Kim, P. M. Proteinsgm: Score-based generative modeling for de novo protein design. *bioRxiv*, 2022.

Lei, T. When attention meets fast recurrence: Training language models with reduced compute. *arXiv preprint arXiv:2102.12459*, 2021.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691, 2021.

Li, T., Pantazes, R. J., and Maranas, C. D. OptMAVEn– a new framework for the de novo design of antibody variable region models targeting specific antigen epitopes. *PlOS one*, 9(8):e105954, 2014.

Li, X. L. and Liang, P. 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, August 2021.

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models. *bioRxiv*, 2022.

Mason, D. M., Friedensohn, S., Weber, C. R., Jordi, C., Wagner, B., Meng, S. M., Ehling, R. A., Bonati, L., Dahinden, J., Gainza, P., et al. Optimization of therapeutic antibodies by predicting antigen specificity from antibody sequence via deep learning. *Nature Biomedical Engineering*, 5(6): 600–612, 2021.

Melnyk, I., Das, P., Chenthamarakshan, V., and Lozano, A. Benchmarking deep generative models for diverse antibody sequence design. *arXiv preprint arXiv:2111.06801*, 2021.

Neekhara, P., Hussain, S., Du, J., Dubnov, S., Koushanfar, F., and McAuley, J. Cross-modal adversarial reprogramming. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2427–2435, 2022.

Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., and Madani, A. ProGen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022.

Olsen, T. H., Boyles, F., and Deane, C. M. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1):141–146, 2022.

Pantazes, R. and Maranas, C. D. OptCDR: a general computational method for the design of antibody complementarity determining regions for targeted epitope binding. *Protein Engineering, Design & Selection*, 23(11):849–858, 2010.

Raybould, M. I. J., Marks, C., Krawczyk, K., Taddese, B., Nowak, J., Lewis, A. P., Bujotzek, A., Shi, J., and Deane, C. M. Five computational developability guidelines for therapeutic antibody profiling. *Proceedings of the National Academy of Sciences*, 116(10):4025–4030, 2019.

Raybould, M. I. J., Kovaltsuk, A., Marks, C., and Deane, C. M. CoV-AbDab: the Coronavirus Antibody Database. *Bioinformatics*, 37(5):734–735, 2021.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118 (15), 2021.

Ruder, S. Recent Advances in Language Model Fine-tuning. `http://ruder.io/recent-advances-lm-fine-tuning`, 2021.

Ruffolo, J. A., Chu, L.-S., Mahajan, S. P., and Gray, J. J. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *bioRxiv*, 2022.

Saka, K., Kakuzaki, T., Metsugi, S., Kashiwagi, D., Yoshida, K., Wada, M., Tsunoda, H., and Teramoto, R. Antibody design using lstm based deep generative model from phage display library for affinity maturation. *Scientific reports*, 11(1):1–13, 2021.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Shin, J.-E., Riesselman, A. J., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., and Marks, D. S. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):1–11, 2021.

Steinegger, M. and Söding, J. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1): 1–8, 2018.

Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., and Kim, P. M. Fast and flexible design of novel proteins using graph neural networks. *BioRxiv*, pp. 868935, 2020.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Syrlybaeva, R. and Strauch, E.-M. Deep learning of protein sequence design of protein-protein interactions. *bioRxiv*, 2022.

Tian, P. and Best, R. B. How many protein sequences fold to a given structure? A coevolutionary analysis. *Biophysical journal*, 113(8):1719–1730, 2017.

Tobias H. Olsen, I. H. M. and Deane, C. M. Ablang: An antibody language model for completing antibody sequences. *bioRxiv*, 2022.

Tsai, Y.-Y., Chen, P.-Y., and Ho, T.-Y. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning*, pp. 9614–9624, 2020.

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

Vinod, R., Chen, P.-Y., and Das, P. Reprogramming language models for molecular representation learning. *arXiv preprint arXiv:2012.03460*, 2020.

Wang, J., Cao, H., Zhang, J. Z., and Qi, Y. Computational protein design with deep learning neural networks. *Scientific reports*, 8(1):1–9, 2018.

Weitzner, B. D., Dunbrack Jr, R. L., and Gray, J. J. The origin of CDR H3 structural diversity. *Structure*, 23(2): 302–311, 2015.

Yang, C.-H. H., Tsai, Y.-Y., and Chen, P.-Y. Voice2Series: Reprogramming acoustic models for time series classification. In *International Conference on Machine Learning*, pp. 11808–11819, 2021.

Yang, K., Jin, W., Swanson, K., Barzilay, R., and Jaakkola, T. Improving molecular design by stochastic iterative target augmentation. In *International Conference on Machine Learning*, pp. 10716–10726, 2020.

Yu, L., Tanwar, D. K., Penha, E. D. S., Wolf, Y. I., Koonin, E. V., and Basu, M. K. Grammar of protein domain architectures. *Proceedings of the National Academy of Sciences*, 116(9):3636–3645, 2019.

Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

A. Related Work on Protein Design

Protein design involves the design of new protein sequences that fold to a desired 3D structure and/or exhibit a specific function. Computational techniques for designing novel and diverse proteins are an active area of research. Physics based methods that rely on energy minimization have been proposed for designing general proteins (Leaver-Fay et al., 2011; Huang et al., 2011), as well as specifically for antibodies (Pantazes & Maranas, 2010; Li et al., 2014; Adolf-Bryfogle et al., 2018), but these are computationally expensive. Recently, generative deep learning techniques like Generative Adversarial Networks (Goodfellow et al., 2020), Variational Autoencoders (Kingma & Welling, 2013), Graph Neural Networks (Scarselli et al., 2008; Gilmer et al., 2017), autoregressive language models (LSTM and Transformer based) (Vaswani et al., 2017), and diffusion based models (Ho et al., 2020) have been used for protein and antibody design (Wang et al., 2018; Akbar et al., 2022; Amimeur et al., 2020; Eguchi et al., 2020; Shin et al., 2021; Kong et al., 2022; Fu & Sun, 2022; Syrlybaeva & Strauch, 2022; Lee & Kim, 2022; Anand & Achim, 2022). Some representative works are discussed below. (Ingraham et al., 2019) and (Cao et al., 2021) proposed a graph and a multimodal transformer based model, respectively, for designing proteins conditioned on the backbone structure/fold. (Karimi et al., 2020), developed a guided conditional Wasserstein Generative Adversarial Networks (gcWGAN) for fold based protein design. Another method that uses GANs to generate a distance matrix representation of proteins from which 3D coordinates can be recovered was proposed by (Anand & Huang, 2018). Variational autoencoder based methods have also been proposed for conditional generation of protein sequences (Greener et al., 2018; Das et al., 2021) and for direct generation of 3D coordinates of immunoglobulin proteins (Eguchi et al., 2020).

Several of the above-mentioned architectures have been extended to the specific problem of antibody design, which is considered challenging due to focus on designing long, variable, and unstructured CDRs. (Melnyk et al., 2021) provides benchmarking of several deep generative models on antibody design. Recently, (Jin et al., 2021) proposed an iterative refinement graph neural network for jointly designing the sequence and 3D structure of the CDR regions of antibodies for improving its properties. A deep generative model that jointly models sequences and structures of CDRs based on diffusion processes and equivariant neural networks has been proposed in (Luo et al., 2022). A geometry-constrained energy-based model has been suggested by (Fu & Sun, 2022).

Other approaches for protein design include modeling it as a constraint satisfaction problem (Strokach et al., 2020), equivariant 3D translation (Kong et al., 2022) and by using combinatorial bayesian optimization (Khan et al., 2022).

B. Overview of Proposed Baseline Models

Figure 5 shows diagrams of the proposed baseline BERT-based infilling models: ProtBert, a specialized model that has been pretrained on millions of protein sequences and EnglishBert, the traditional English language model, where we replaced word embeddings with new learnable amino acid embeddings. Similar as our main proposed method, ReprogBert, these two models are sequence-only methods and they use maskings to infill the regions of interest.

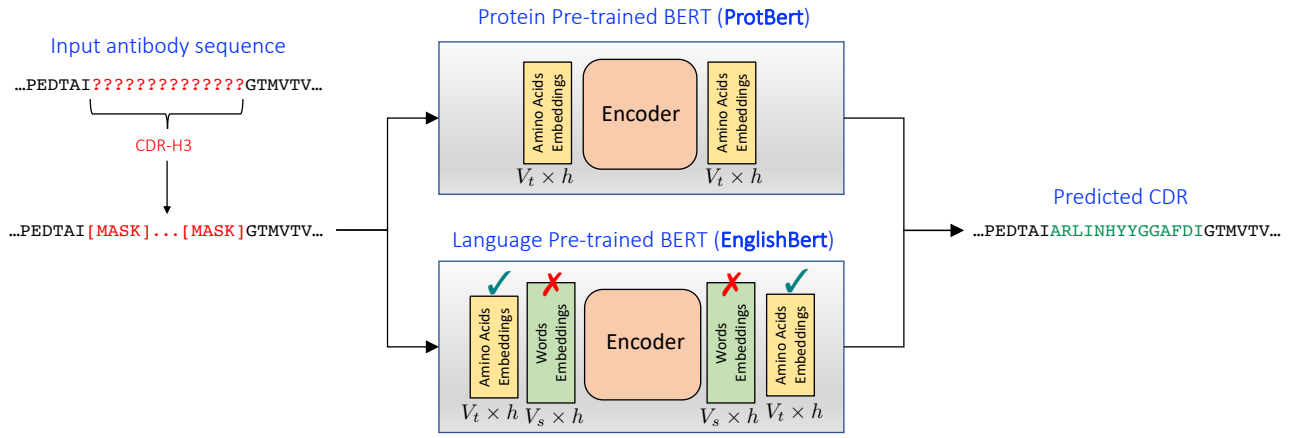

Figure 5. Baseline methods proposed for protein sequence infilling. Given an input antibody sequence, where part of the amino acids is missing (e.g., CDR-H3), the goal is to infill them using information from the rest of the protein. The infilling problem is formulated similar to the masked-language modeling task, where the missing amino acids are marked with a token ⟨MASK⟩ and the model generates amino acids token to infill them. These are sequence-only methods and do not rely on any structure information during generation process. The top diagram shows *ProtBert*, the BERT model that has been pretrained on the protein sequences and therefore can be applied to the protein infilling task as is (the entire model is still fine-tuned on the downstream infilling task). The bottom diagram shows traditional English language BERT model (*EnglishBert*), whose incompatible word embeddings ($V_s \times h$, V_s is the number of language tokens, h - latent model dimension) are swapped with the trainable amino acid embeddings ($V_t \times h$, V_t is the number of amino acid tokens). The full model is then fine-tuned on the infilling dataset.

C. Model Architecture and Training

In Table 11 we present the architectural details of our BERT-based models for the protein sequence infilling, while Table 12 shows the settings used for model training.

Model	Number of parameters	Number of layers	Hidden layer size	Number of heads	Vocab size	Pretraining Data	Reference
ProtBert	420M	30	1024	16	30	BFD100 (572 GB, 2 bil proteins) Uniref100 (150 GB, 216 mil proteins)	(Devlin et al., 2018)
EnglishBert / ReprogBert (based on HF bert-base-uncased)	110M	12	768	12	30522 (english) 30 (protein)	English Wikipedia (40 GB, 6.5 mil sentences) BookCorpus (6 GB, 74 mil sentences)	(Elnaggar et al., 2020)

Table 11. Architectural details of the BERT-based model for protein sequence infilling. Note that for ReprogBert the number of trainable parameters is defined by the two $\mathbb{R}^{30522 \times 30}$ matrices.

Learning rate	Batch size	Optimizer
$1e^{-5}$	32	Adam

Table 12. Training details for ProtBert, EnglishBert and ReprogBert. For example, for SabDab dataset to reach the best performance it took 5 hours for ReprogBert, 6 hours for EnglishBert and 14 hours for ProtBert, which is equivalent to approximately 1800 epochs (134 minibatch iterations per epoch). We trained all models on a single A100 40GB GPU. Average inference time per protein sequence is 0.02 seconds for ProtBert, and 0.008 seconds for ReprogBert and EnglishBert (as measured on the test set of SabDab for CDR-H3 infilling). For reference, the average inference time for RefineGNN is 0.004 seconds, which is comparable to our ReprogBert.

D. Ablation on Data

In Table 13 we show an ablation results on the effect of training data size on model performance.

SabDab-H3				
	Training data fraction	PPL-ProGen	AAR	DIV
ProtBert	1.0	6.8	41.5	14.5
	0.8	6.7	41.3	13.1
	0.6	6.6	40.9	15.9
	0.4	6.4	40.5	18.9
	0.2	6.6	40.3	18.4
EnglishBert	1.0	5.9	35.9	59.8
	0.8	5.9	35.1	57.9
	0.6	6.5	34.2	59.6
	0.4	6.4	33.6	61.4
	0.2	6.5	33.1	63.5
ReprogBert	1.0	6.0	32.6	67.4
	0.8	5.9	32.1	67.6
	0.6	6.1	31.6	68.2
	0.4	6.3	30.8	69.5
	0.2	6.5	29.9	70.7

Table 13. Ablation results on the effect of training data size on model performance. The fractions 1.0, 0.8, 0.6, 0.4 and 0.2 representing progressively smaller subsets of the original SabDab training dataset. It can be seen that as the size of training data drops, the recovery rate also decreases, while the diversity increases (this is expected as now the generated sequences are less accurate). However, for ProtBert, the decrease is slower, likely due to this model being pretrained on large protein dataset, thus retaining its prediction capacity.

E. Ablation on Models

In Tables 14, 15, and 16 we show ablation results on the effect of model sizes and pre-training on the performance of EnglishBert and ReprogBert models when trained and tested on SabDab dataset. One observation we can make is that the larger pre-trained model (bert-large-uncased, 340M) results in lower AAR and higher DIV as compared to using smaller pre-trained bert-base-uncased (110M). This trend is more pronounced for EnglishBert, which sees more abrupt drops in AAR and significant increase in DIV for CDR-H1 and CDR-H2. This is expected as the model becomes less accurate and more random in generating the CDRs. On the other hand, ReprogBert is more stable and we see a smaller change in recovery and diversity. For CDR-H3 both EnglishBert and ReprogBert have similar drop in accuracy accompanied with smaller increase in diversity. We can conclude that increasing the model capacity did not improve performance, likely due to the limited size of available training data. On the other hand, using pre-trained language models is still beneficial as compared to starting from scratch (third column in the Tables).

SabDab CDR-H1									
	Base (110M)			Large (340M)			Scratch (110M)		
	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV
EnglishBert	3.7	63.6	5.8	7.5	43.8	63.4	15.1	38.3	71.9
ReprogBert	3.3	56.0	29.1	5.2	51.5	39.4	4.5	48.7	45.5

Table 14. Ablation results on the effect of the model size and pre-training on the performance of EnglishBert and ReprogBert models for CDR-H1.

	SabDab CDR-H2								
	Base (110M)			Large (340M)			Scratch (110M)		
	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV
EnglishBert	4.0	59.1	7.7	8.9	40.5	68.7	10.3	38.4	72.7
ReprogBert	3.9	53.0	37.9	8.5	50.9	43.4	13.8	43.6	62.8

Table 15. Ablation results on the effect of the model size and pre-training on the performance of EnglishBert and ReprogBert models for CDR-H2.

	SabDab CDR-H3								
	Base (110M)			Large (340M)			Scratch (110M)		
	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV	PPL-ProGen	AAR	DIV
EnglishBert	5.9	35.6	59.8	5.5	32.2	68.2	6.9	20.1	85.1
ReprogBert	5.4	32.6	67.4	5.6	29.7	70.9	7.0	20.3	84.8

Table 16. Ablation results on the effect of the model size and pre-training on the performance of EnglishBert and ReprogBert models for CDR-H3.

F. Antibody Developability Prediction Task

Here we present additional experimental results on antibody developability. We use the web server from (https://opig.stats.ox.ac.uk/webapps/newsabdab/sabpred/tap) for developability prediction for 35 randomly selected sequences from our ReprogBert-based generations on the SabDab database. Only 2 of them (5.7%) show red flag for at least one of the five developability metrics, indicating a previously unobserved value for that property, as described in (Raybould et al., 2019). This analysis indicates that the generated antibody sequences by ReprogBert do not pose any significant developability concern.

G. Protein-Protein Docking

For this task we have predicted the structure of the ReprogBert-designed antibody sequence for COVID use-case with Spike receptor binding domain from PDB id: 7l7d by using AbAdapt web server (https://sysimm.org/abadapt/). Two such docked structures are shown as examples, which represent the cluster ($size > 1$ structure) with the best docking score. The antibody residues are colored according to the epitope probabilities using an RGB (high to low) scale. Fig. 6 shows a socked structure of BD55-6478 with Spike receptor binding domain (from pdb id: 7l7d), and Fig. 7 shows a docked structure of BD56-124 with Spike receptor binding domain (from pdb id: 7l7d).

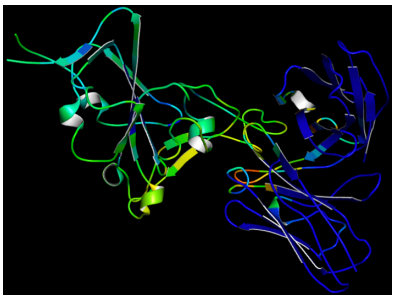

Figure 6. Docked structure of BD55-6478 with Spike receptor binding domain (from pdb id: 7l7d)

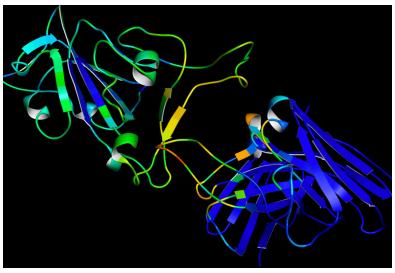

Figure 7. Docked structure of BD55-6478 with Spike receptor binding domain (from pdb id: 7l7d)

H. Examining Amino Acid Embeddings and Mappings

In this section we examine and visualize the embeddings of amino acids and see if there are any naturally forming clustering present. We also examine the mappings between the protein space of amino acids and the space of English tokens of our ReprogBert model. For this, we first review the linear projection method and examine the corresponding embeddings, then we show that it is not easy to use it to visualize the mappings, and propose the alternative based on cross-attention mechanism.

H.1. Projection

Recall from Section 2 that we reprogram EnglishBert model by introducing two linear projection matrices

$$\theta \in \mathbb{R}^{|V_t| \times |V_s|} \tag{10}$$

$$\gamma \in \mathbb{R}^{|V_s| \times |V_t|} \tag{11}$$

to project the target protein domain x_t to source English token domain x_s, and then similarly reverse the mapping of the output, we get, respectively:

$$x_s = x_t \theta \tag{12}$$

$$y_t = \gamma y_s. \tag{13}$$

In particular, focusing on the input projection matrix θ and treating x_t as a one-hot sequence representation of the amino acids of length N, i.e., $x_t \in \mathbb{R}^{N \times |V_t|}$, then the sequence representation in the English token domain becomes $x_s \in \mathbb{R}^{N \times |V_s|}$. Then, this representation is projected onto the embedding matrix of the English Bert model $E \in \mathbb{R}^{|V_s| \times d}$ (d is the hidden dimension of English Bert):

$$x_s^E = x_s E, \tag{14}$$

and continue the usual processing through the transformer layers and blocks.

From (14), the amino acid embeddings $E_{aa} \in \mathbb{R}^{|V_t| \times d}$ can be defined as

$$E_{aa} = \theta E. \tag{15}$$

We use multi-dimensional scaling (MDS) algorithm to project all the 20 amino acids E_{aa} from d-dimensional space into 2D. The resulting scatter plot is shown in Fig. 8. Each of the four plot shows one of the ways to group amino acids based on various biological properties. The top left plot shows the hydrophobic (amino acids encircled by the grey area) versus hydrophilic (all the remaining amino acids). The top right plot shows the clustering based on size: the larger amino acids are highlighted by the grey area, while the amino acids outside the region have the smaller size. Bottom left shows the group of aromatic amino acids. Finally, the bottom right plot shows the cluster of negatively and positively charged amino acids. As can be seen, the learned embedding matrix for amino acids encodes meaningfull grouppings of the amino acids, with the amino acids from the same biological group collocated close to each other. This signifies that ReprogBert is able to properly project pretrained English word token embeddings and create new protein embeddings with meaningful biological properties.

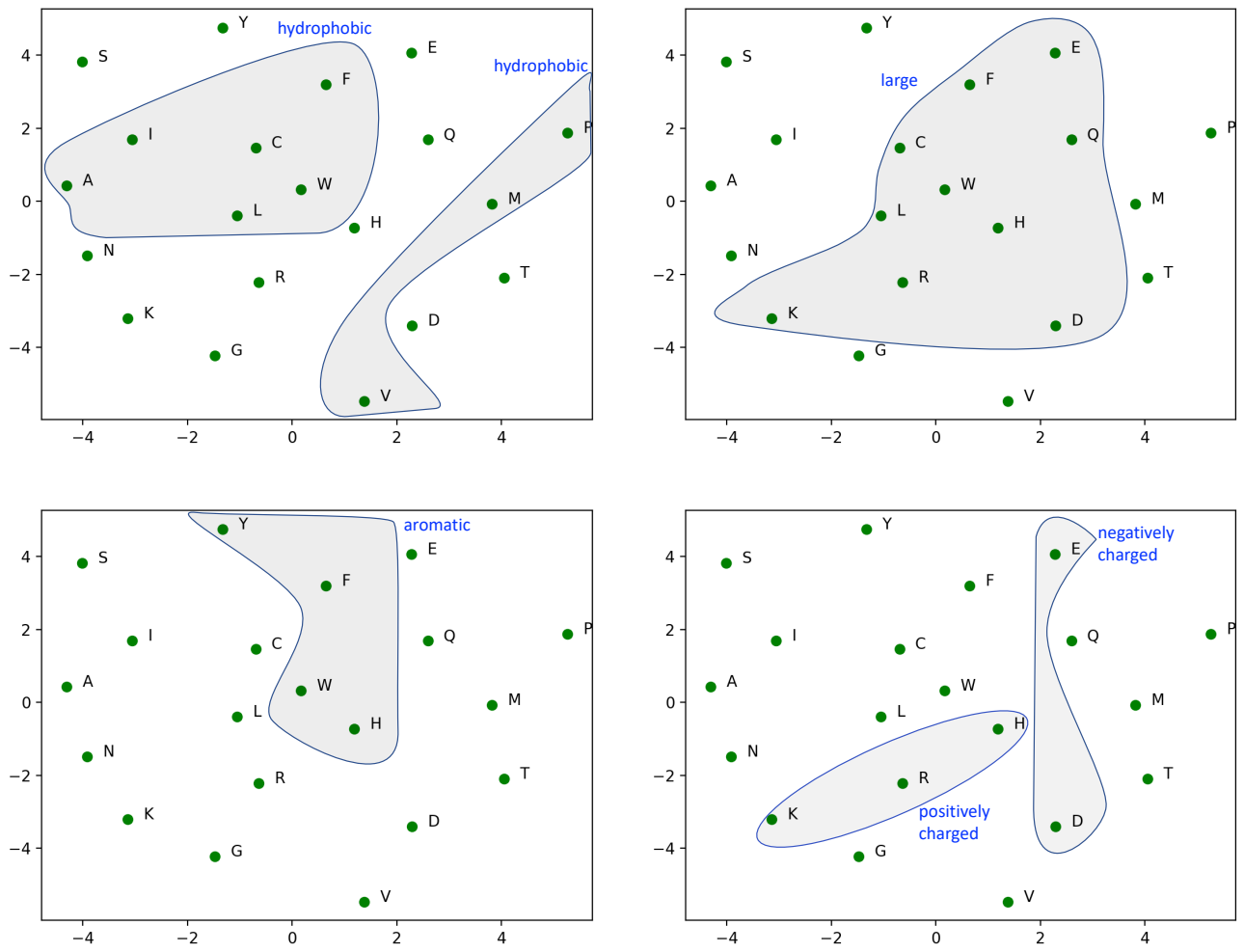

Figure 8. Projection of 20 amino acids, defined by embedding matrix E_{aa} (see equation (15)), onto the 2D plane using multi-dimensional scaling algorithm. Each of the four plot shows one of the ways to group amino acids based on various biological properties. The top left plot shows the hydrophobic (amino acids encircled by the grey area) versus hydrophilic (all the remaining amino acids). The top right plot shows the clustering based on size: the larger amino acids are highlighted by the grey area, while the amino acids outside the region have the smaller size. Bottom left shows the group of aromatic amino acids. The bottom right plot shows the cluster of negatively and positively charged amino acids.

Since x_s is usually a dense vector, its projection onto embedding vector has limited interpretability in that all rows of E are mixed during the projection. To better understand and visualize the mapping between amino acids and English tokens, we propose to replace the above projection with cross-attention mechanism.

H.2. Cross-Attention

For this, we redefine θ to be the amino acids embedding matrix and keep the English tokens embedding matrix as is:

$$\theta \in \mathbb{R}^{|V_t| \times d} \tag{16}$$

$$E \in \mathbb{R}^{|V_s| \times d}, \tag{17}$$

and define the multi-head cross attention mechanism between θ and E. We split d (in our case $d = 768$) into h parts of size $d_i = d/h$ each, resulting in

$$\theta = [\theta_1, \ldots, \theta_h], \quad \text{for } \theta_i \in \mathbb{R}^{|V_t| \times d_i} \tag{18}$$

$$E = [E_1, \ldots, E_h], \quad \text{for } E_i \in \mathbb{R}^{|V_s| \times d_i}, \tag{19}$$

and also define d_i learnable projection matrices

$$W_i^Q \in \mathbb{R}^{d_i \times d_i} \tag{20}$$
$$W_i^K \in \mathbb{R}^{d_i \times d_i} \tag{21}$$
$$W_i^V \in \mathbb{R}^{d_i \times d_i}. \tag{22}$$

The projection of θ and E gives us the key, query and value matrices:

$$Q_i = \theta_i W_i^Q \tag{23}$$
$$K_i = E_i W_i^K \tag{24}$$
$$V_i = E_i W_i^V, \tag{25}$$

which are then projected to obtain the cross-attention

$$\bar{E} = \left[\bar{E}_1, \ldots, \bar{E}_h\right], \quad \text{where } \bar{E} \in \mathbb{R}^{|V_t| \times d} \tag{26}$$
$$\bar{E}_i = hardmax\left(\frac{Q_i K_i^T}{\sqrt{d_i}}\right) V_i, \quad \text{where } \bar{E}_i \in \mathbb{R}^{|V_t| \times d_i}, \tag{27}$$

where to compute the attention weights, we replaced the traditional softmax operation with hardmax, which converts the probabilities into one-hot prepresentation

$$A_i = hardmax\left(\frac{Q_i K_i^T}{\sqrt{d_i}}\right) \in \mathbb{R}^{|V_t| \times |V_s|}, \tag{28}$$

where each row is all zeros except a single one in one of the columns. In practice, to enable differentiation, we use straight-through Gumbel-softmax to implement the differentiable hardmax operation.

Note that by using one-hot attention enables us to see exactly which of the amino acids is associated with which English token, therefore improve mapping visualization. However, since we use multiple heads, the mapping is not one to one but can be one to many, where a single amino acid is mapped into multiple English tokens due to the concatenation of resulting sub-embedding matrices:

$$\bar{E} = \left[\bar{E}_1, \ldots, \bar{E}_h\right]. \tag{29}$$

Now the the input amino acids sequence can be directly used to as input to the model:

$$x_s^{\bar{E}} = x_s \bar{E}, \tag{30}$$

and continue the usual processing through the transformer layers and blocks. Note that similar derivations can be constructed to derive the cross-attention for γ, mapping amino acid space into the English token space.

To summarize, the projection method in Section H.1 optimizes only two learnable matrices θ and γ, while keeping the rest of the ReprogBert parameters frozen. On the other hand, in the cross-attention approach, additionally to θ and γ, we also learn h attention matrices W_i^Q, W_i^K, and W_i^V.

The comparison between the projection and cross-attention approaches is shown in Table 17.

As, can be seen the cross-attention with 4 heads or more performs well, with 8 and 12 heads matching the projection method closely. In Figure 9 we present the map for the case of cross-attention with 4 heads for the ease of visualization. Here, each amino acid is mapped into some combination of 4 word tokens, where each of 4-token combindations is unique. The set of all the word tokens (13 of them) which were used in the map are shown in Figure 10.

	SabDab CDR-H3		
	PPL-ProGen	AAR	DIV
ReprogBert (projection)	5.4	32.6	67.4
ReprogBert (cross-attention, 1 head)	94.5	5.2	94.1
ReprogBert (cross-attention, 2 heads)	94.1	5.0	93.2
ReprogBert (cross-attention, 4 heads)	7.0	26.8	79.5
ReprogBert (cross-attention, 6 heads)	5.8	30.2	68.2
ReprogBert (cross-attention, 8 heads)	5.9	31.6	67.9
ReprogBert (cross-attention, 12 heads)	5.9	31.1	66.7

Table 17. Performance comparison between various mappings in ReprogBert as trained and tested on SabDab CDR-H3 dataset. Cross-attention with a single or 2 heads is not performing well, while larger number of heads improves the performance. 8 heads cross-attention matches the performance of linear projection (first row).

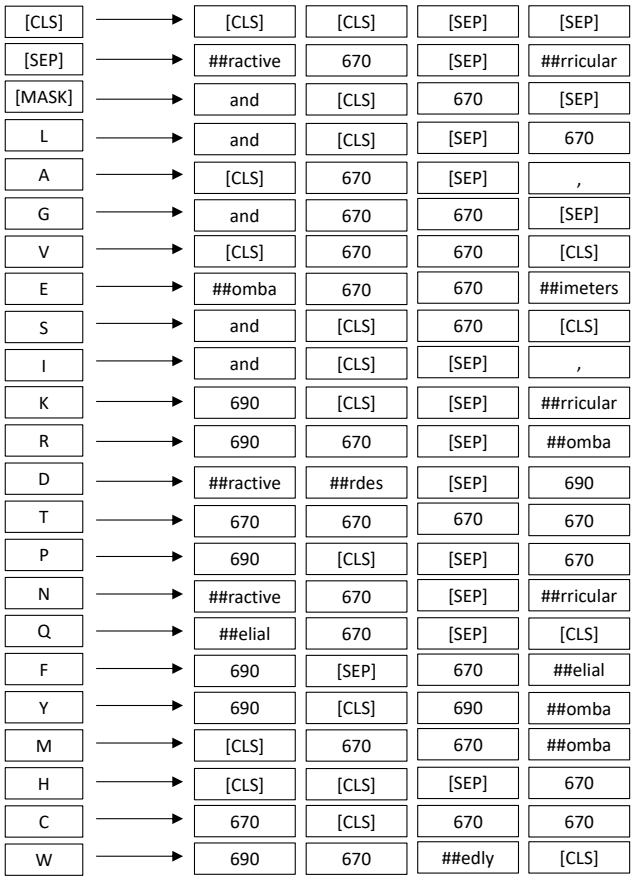

Figure 9. The mapping (A_i as defined in equation (28)) between amino acid vocabulary and the English word tokens for ReprogBert model with cross-attention using 4 heads. As can be seen, each amino acid is mapped into some combination of 4 word tokens, where each of 4-token combindations is unique. It appears there is no particular semantic meaning in this map, rather it follows some statistical relationships the model optimized during the training. At the same time, it is interesting to observe that to get a good performance the model had to use at least 4 heads (corresponding to 4 word tokens per amino acid), while using fewer, such as a single or two words per token, did not produce satisfactory performance.

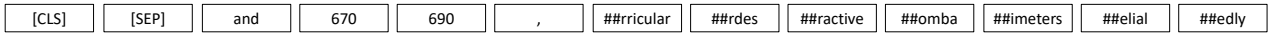

Figure 10. All the word tokens used in the map of Figure 9. There are in total 13 word tokens.

I. Recovery and Diversity Metrics

Figures 11 and 12 show additional visualizations of the recovery and diversity metrics for CDR-H3 across different methods.

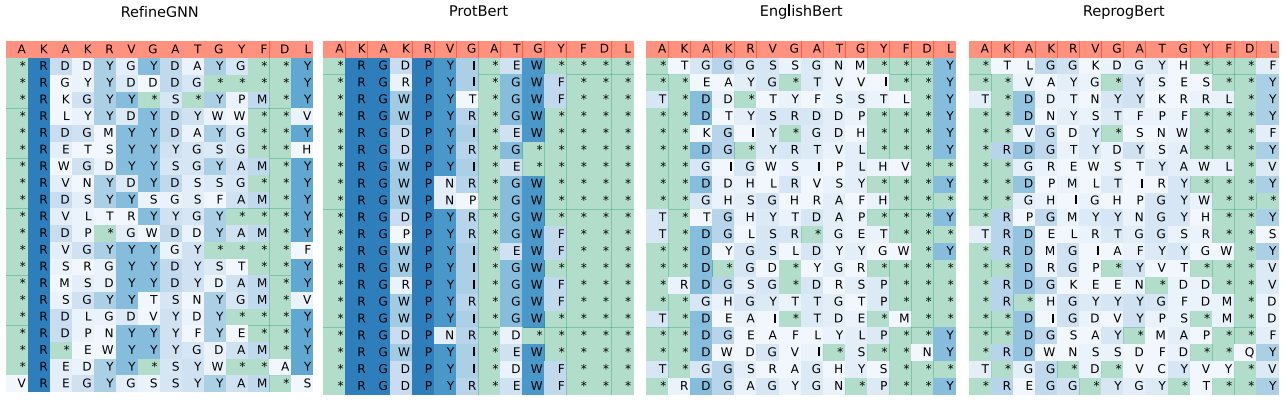

Figure 11. Visualization of the recovery and diversity metrics for CDR-H3 (PDB ID 2r56) across different models. The top red line shows the ground truth CDR-H3 sequence, while the next lines show the generated CDR-H3 by each of the model. The green cell with the star symbol represents the same amino acid as in the ground truth, while the blue cell shows new and different generated residues. The shade of the blue cell represents the frequency of the amino acid in that column. We see that ReprogBert has highest diversity represented by the largest number of light blue cells, at the same time ProtBert has most green cells (highest AAR), but also many dark blue cells (low diversity). It can also be seen that RefineGNN has lower diversity and lower recovery, as compared to ReprogBert.

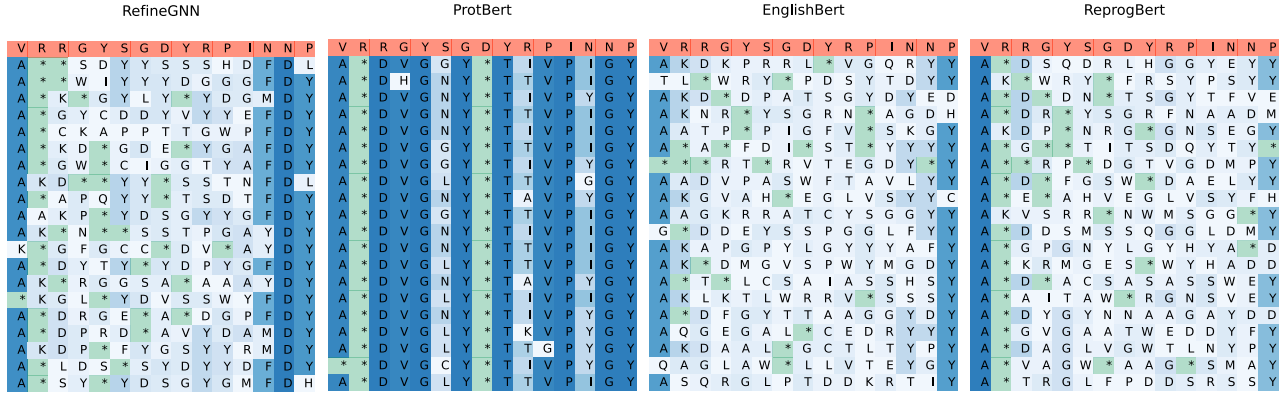

Figure 12. Visualization of the recovery and diversity metrics for CDR-H3 (PDB ID 5y7z) across different models. The top red line shows the ground truth CDR-H3 sequence, while the next lines show the generated CDR-H3 by each of the model. The green cell with the star symbol represents the same amino acid as in the ground truth, while the blue cell shows new and different generated residues. The shade of the blue cell represents the frequency of the amino acid in that column. We see that ReprogBert has highest diversity represented by the largest number of light blue cells, at the same time ProtBert has most green cells (highest AAR), but also many dark blue cells (low diversity). It can also be seen that RefineGNN has lower diversity and lower recovery, as compared to ReprogBert.

J. Physicochemical Property Comparison

Finally, in Figure 13 we show 2D kernel density plot as a function of isoelectric point and length of generated CDR-H3 sequences on the test set of SabDab dataset.

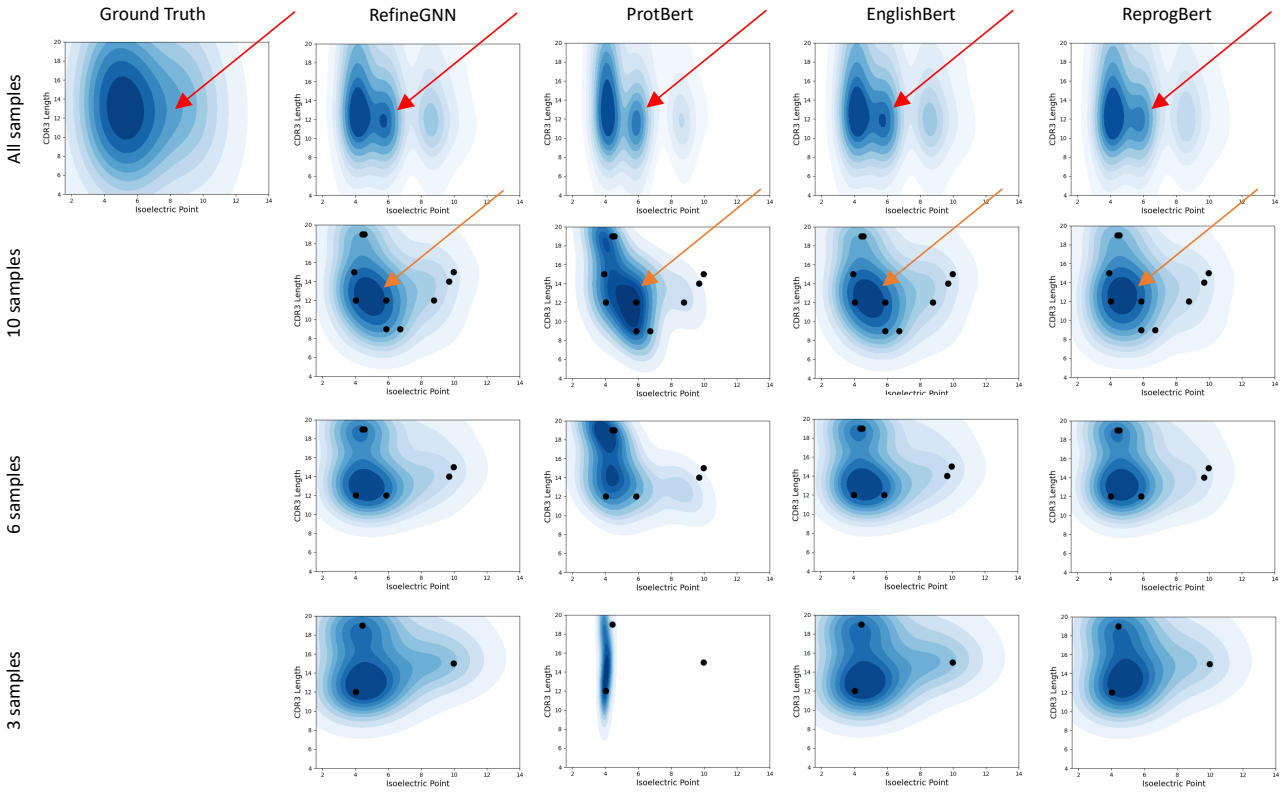

Figure 13. 2D kernel density plot as a function of isoelectric point, the pH at which a particular molecule carries no net electrical charge, and length of generated CDR-H3 sequences on the test set of SabDab dataset. Black dots indicate the ground truth CDR-H3. The top row shows the density of all the sequences, while the following rows show the density for 10, 6 and 3 samples. It can be seen that the ground truth density of all the sequnces (top left corner) has one pronounced peak (for the CDR3 length 13 and pH 5) and another smaller increase of density marked with red arrow. Comparing this region across other models, we see that ReprogBert has the closes resemblance to the ground truth, while others place too much weight there. The second row from the top shows the density for 10 protein sequences, where visual inspection of the region marked with orange arrow reveals that ReprogBert has closest similarity to the ground truth based on the distribution and orientation of the highly dense region, while for other methods theshape of this region is tilted and a second minimum appears.