Generative Antibody Design for Complementary Chain Pairing Sequences through Encoder-Decoder Language Model

Simon K.S. Chu[*]
University of California Davis
Davis, CA 95616
kschu@ucdavis.edu

Kathy Y. Wei[†]
Therapeutic Discovery, Amgen Research, Amgen Inc.
South San Francisco, CA 94080
kywei@alumni.stanford.edu

Abstract

Current protein language models (pLMs) predominantly focus on single-chain protein sequences and often have not accounted for constraints on generative design imposed by protein-protein interactions. To address this gap, we present paired Antibody T5 (pAbT5), an encoder-decoder model to generate complementary heavy or light chain from its pairing partner. We show that our model respects conservation in framework regions and variability in hypervariable domains, demonstrated by agreement with sequence alignment and variable-length CDR loops. We also show that our model captures chain pairing preferences through the recovery of ground-truth chain type and gene families. Our results showcase the potential of pAbT5 in generative antibody design, incorporating biological constraints from chain pairing preferences.

1 Introduction

Transformer-based protein language models (pLMs) have begun to find utility across a range of applications in the field. Remarkably, even when pretrained solely on sequence databases, these models have demonstrated the ability to aid in protein structure prediction [1, 2] and a host of downstream tasks including function and secondary structure annotations [3–7]. Furthermore, they have shown promise in the area of *de novo* protein design, proving to be useful in efforts ranging from point mutation design to full-sequence generation [8–14]. By leveraging the evolutionary information contained in sequence databases, pLMs offer a pathway to understanding and designing protein sequences through a language modeling approach.

Most pLMs are designed for single-chain sequences only. However, many biological contexts involve protein-protein interactions where multiple chains interact simultaneously. For instance, antibodies consist of paired heavy and light chains. Modeling heavy and light chains independently is inadequate to reflect their heterodimeric nature and sacrifices their co-evolutionary information. Understanding antibody chain pairing has the potential to generate partner sequences given an existing heavy or light chain target.

To address this gap, we present paired Antibody T5 (pAbT5) to generate antibody sequences conditioned on their chain pairing partner in an encoder-decoder architecture. To summarize,

- We modeled antibody chain pairing as a conditional protein design problem through T5 architecture.

[*]Work done as an intern at Therapeutic Discovery, Amgen Research, Amgen Inc.
[†]Currently CSO and cofounder at 310 AI Inc.

NeurIPS 2023 Generative AI and Biology Workshop.

- We show that our model generates antibody sequences respecting conservation in framework region and variability in hypervariable domains.
- We show that our generated sequences capture chain pairing preferences through the recovery of ground-truth chain type and gene families.

2 Related Work

Prior works in generative antibody language models usually are based on either causal language models or denoising neural networks. Nijkamp et al. [13] built a decoder-only model on single-chain antibody sequences. Shuai et al. [15] extended the framework to conditional generation with species and chain type prefix tokens. Denoising network from Frey et al. [14] generates variable-length paired antibody sequences by introducing gap tokens. Distinct from language models, inverse folding models are capable of generating multiple-chain sequences based on structural inputs [16, 17].

3 Methods

3.1 Model and Optimization

We approach the antibody chain pairing problem under a sequence-to-sequence generation framework. We use the term forward-translation to describe light-to-heavy-chain generation and back-translation for the reciprocal process. Notably, we do not specify the translation direction, nor do we include any gap or prefix tokens relating to the input or target chain type, species, or gene families in our model. The model is fine-tuned from ProtT5-XL-UniRef50, which has a T5 architecture [5].

To optimize our model, we adhere to the ProtT5-XL pretraining scheme utilizing a local batch size of 8 and a global batch size of 2048. We kept the encoder weights frozen and fine-tuned only on the decoder and observed better encoder representation on sequences compared to fine-tuning the whole model. We used a learning rate of 5e-5 without weight decay in AdaFactor optimizer with a gradient clipping of 1 and a patience of 5 epochs on validation loss for two days on eight A100 GPUs. The implementation is on PyTorch under HuggingFace framework [18, 19].

3.2 Dataset

We sourced approximately 160k pairs of antibody VH and VL sequences from the Observed Antibody Space (OAS) database [20]. Leveraging the framework of forward- and back-translations, we represented each bi-directional pairing through two uni-direction translations. This yielded a dataset of roughly 321k translation samples derived from 239k distinct sequences from humans, rats, and mice.

In the context of the protein-protein interaction network in the OAS dataset, edge-based splitting serves as an intuitive method for data partitioning. An alternative approach is node-based partitioning, where all edges linked to training nodes are incorporated into the training set, leaving the rest for testing. We employ an exclusive node split strategy, reserving specific nodes and their related edges solely for testing to rigorously evaluate the model's generalization to unseen sequences and pairings (Figure A.1). Consequently, our dataset is partitioned into a roughly 90-5-5 distribution, resulting in 260k training, 828 validation, and 802 test translations.

4 Results

4.1 Sequence Generation Aligns with Conserved and Variable Domains in Antibodies

Antibodies display significant diversity in their hypervariable domains to ensure specificity in antigen binding. Both the light and heavy chains possess three loop structures, known as the CDR loops. While these loops are highly variable, other regions, termed framework regions, remain relatively conserved. Of all the CDR loops, the third loop on the heavy chain (CDRH3) exhibits the highest variability. In this section, we evaluate whether our model successfully recognizes and reproduces these distinct patterns during next-word prediction and sequence generation.

In Figure 1, we compare the probability from next-word prediction against conservation from alignment analysis. The model demonstrates higher confidence in the more conserved framework region of the heavy chain target and displays increased uncertainty in the variable CDR loops. To further examine its ability to generate realistic sequences, we align the observed and generated sequences for a random heavy-light chain pairing from the test set. Notably, generated sequences often exhibit greater variability than next-word probabilities, potentially due to the cascading effect during iterative sampling. These sequences might also originate from different gene loci or families than the target sequences. This analysis highlights the model's ability to generate variable-length CDR (H3) loops while preserving patterns in framework regions. On average, generated sequences maintain approximately 60% whole-sequence identity with target sequences. This suggests our model effectively balances capturing antibody pairing patterns and creating novel sequences. For a detailed analysis of sequence identities and lengths by region, see Tables A.4, A.5 and A.6. Comprehensive alignment profiles for both heavy and light chains, along with four other random output samples from the test set, can be found in Figures A.18, A.19, A.20, and A.21.

Figure 1: Comparison of observed and model-derived alignment profiles on heavy chain across framework regions (FR) and CDR loops. The first and second rows pair the next-word probability under teacher-forcing with sequence conservation from alignment to UniRef90 [21]. The third and fourth rows provide a side-by-side view of global alignments between generated sequences and their corresponding observed sequences. The reciprocal analysis on the reverse direction can be found in Figure A.18.

Beyond assessing alignment profiles, we further validate our model predictions by superimposing these results on both predicted structures by DeepAb [22] and known experimental structures. Indeed, the generated heavy and light chains exhibit structurally consistent framework regions while emphasizing variations in the CDR loops, as illustrated in Figure A.16. With the interest to evaluate on unseen experimental structures, we analyzed three antibodies bound to the SARS-CoV-2 spike protein from RCSB database [23, 24]. Figure 2 demonstrates that the CDR loops remain the most entropic regions across all three antibody structures.

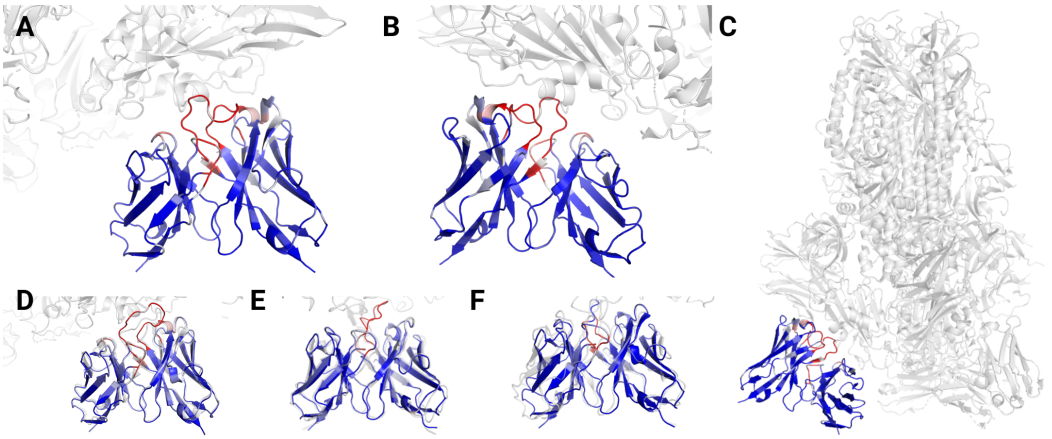


Figure 2: Visualization of next-word prediction entropy for antibodies bound to the SARS-CoV-2 spike protein. Blue denotes low entropy regions, while red represents areas of high entropy. (A and B) Offer front and back views of the 6WPS PDB structure. (C) Provides an overview of the entire 6WPS structure. (D, E, and F) depict structures from 6WPT, 7TB8 chains D and E, and 7TB8 chains H and I, respectively.

4.2 Conditional Generation Recovers Pairing Sequences

The human immune system can recognize a vast array of antigens by generating a diverse repertoire of antibodies through gene rearrangement. The transcription of each antibody sequence is driven by the combination of C, V, and J genes, with an additional D gene specifically for the heavy chain. These genes are stored within chromosome gene loci, specifically H, λ, and κ, and each of these genes corresponds to a segment within the complete antibody sequence. The recombination of VDJ gene families allows for an impressive array of heavy-light chain pairings, estimated at around 10^6 combinations, which are further amplified by somatic mutations [25]

To benchmark our generative model against the current state-of-the-art, we evaluate the percentage of generated sequences sharing the same chain type, gene loci, V, and J gene families as the pairing target. We assess our model against ProGen2-OAS and IgLM, two publicly available state-of-the-art antibody language models. ProGen2-OAS is a decoder-only LM trained on unpaired antibody sequences, and as such, it's not inherently designed to understand antibody pairing [13]. Similarly, IgLM, while focusing on conditions of species and chain type by appending tag tokens at sequence starts, doesn't have an inherent design for pairing comprehension [15]. For ProGen2-OAS, pairing sequences are generated unconditionally. In the IgLM scenario, we provide chain and species tags, assuming the heavy chain must pair with the light chain and both chains belong to the same species. Additionally, we introduce a baseline of selecting a sequence at random from the test set population and another baseline of selecting the pairing partner of the closest sequence from the validation set, termed as *population sampling* and *closest sequence*.

In Table 1, we present a comparison of the percentage of generated sequences that align with the target across various attributes. pAbT5 consistently demonstrates superior performance compared to current state-of-the-art models and baselines. Our model's efficacy significantly surpasses that of *population sampling* and *closest sequence*, suggesting that pAbT5's target recovery is not merely from exploiting dataset biases or memorization. We highlight the importance of the encoder in the T5 architecture by removing cross-attention and retraining on the decoder-only model, which results in a similar performance to *population sampling*. It's notable that ProGen2-OAS exhibits a marked preference for generating heavy chain sequences, aligning with observations from the unpaired OAS dataset [20]. Nijkamp et al. [13] assessed their model by starting sequence generation with the first few tokens. Contrarily, we decided against providing these initial tokens to ensure no possible clues about target gene loci or families were given, especially when these details aren't evident from the chain type of the pairing partner alone. Even when provided with chain type and species tags, IgLM doesn't quite match the performance of our model. pAbT5 sets a new benchmark in most areas, with the exception being gene families with smaller sample sizes, as illustrated in Figure A.4 and A.5.

Percentage of generated sequences sharing the same attributes with target

	Chain type	Gene loci	V gene family	J gene family
Population sampling	0.50 (2163)	0.38 (1644)	0.12 (513)	0.09 (394)
Closest sequence	**1.00** (4300)	**0.76** (3290)	0.17 (750)	0.04 (180)
ProGen2-OAS	0.50 (2171)	0.50 (2140)	0.12 (500)	0.01 (50)
IgLM	**1.00** (4320)	**0.76** (3280)	0.18 (763)	0.03 (133)
Our method (decoder-only)	0.50 (2165)	0.35 (1506)	0.07 (319)	0.09 (400)
Our method (pAbT5)	**1.00** (4320)	**0.78** (3373)	**0.25** (1066)	**0.21** (896)

Table 1: Percentage of generated sequences in the human antibody test set that match the target sequence's chain type, gene loci, and V and J gene families. We compare our model with a random sequence from the population (population sampling), the pairing partner of the nearest sequence from the validation set (closest sequence), ProGen2-OAS [13], IgLM [15], and a decoder-only T5 model by removing the encoder and cross-attention.

5 Conclusion

In this study, we introduced and evaluated pAbT5, demonstrating its efficacy in capturing intricate antibody pairing patterns and generating chain pairing sequences with notable precision relative to target attributes. Its performance, when compared to existing models, suggests its utility as a valuable tool in advancing antibody research and therapeutic exploration.

6 Acknowledgements

We give our special thanks to Ai Ching Lim and Christy Tinberg for their generous support of this project. We thank George Seegan for language model discussion. We thank Yi Zheng, Danyang Gong, and Austin Rice for helpful discussion on gene families and applications in antibodies. We thank Grant Keller for introducing ANARCI.

References

[1] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan Dos, Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, Alexander Rives, and Meta Ai. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. doi: 10.1101/2022.07.20.500902. URL `https://www.biorxiv.org/content/early/2022/10/31/2022.07.20.500902`.

[2] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022. doi: 10.1101/2022.07.21.500999. URL `http://biorxiv.org/content/early/2022/07/22/2022.07.21.500999.abstract`.

[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, 12 2019. ISSN 15487105. doi: 10.1038/s41592-019-0598-1.

[4] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating Protein Transfer Learning with TAPE. *Advances in Neural Information Processing Systems*, 32, 2019. URL `https://github.com/songlab-cal/tape`.

[5] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Learning. *bioRxiv*, 14(8), 2021. URL `https://www.biorxiv.org/content/early/2021/05/04/2020.07.12.199554`.

[6] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 4 2022. ISSN 14602059. doi: 10.1093/bioinformatics/btac020.

[7] 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. doi: 10.1101/2022.08.16.504181. URL `https://www.biorxiv.org/content/early/2022/08/17/2022.08.16.504181`.

[8] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, Rob Fergus, Joshua Meier, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 118(15):e2016239118, 4 2019. doi: 10.1101/622803. URL `http://www.pnas.org/content/118/15/e2016239118.abstract`.

[9] John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in Neural Information Processing Systems*, pages 15820–15831, 2019.

[10] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. ProGen: Language Modeling for Protein Generation. *arXiv*, 3 2020. URL http://arxiv.org/abs/2004.03497.

[11] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:1–28, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f51338d736f95dd42427296047067694-Abstract.html.

[12] Vladimir Gligorijević, Daniel Berenberg, Stephen Ra, Andrew Watkins, Simon Kelow, Kyunghyun Cho, and Richard Bonneau. Function-guided protein design by deep manifold sampling. *bioRxiv*, 2021. doi: 10.1101/2021.12.22.473759. URL https://doi.org/10.1101/2021.12.22.473759.

[13] Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the Boundaries of Protein Language Models. *arXiv*, 6 2022. URL http://arxiv.org/abs/2206.13517.

[14] Nathan C. Frey, Daniel Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, Andreas Loukas, Vladimir Gligorijevic, and Saeed Saremi. Protein Discovery with Discrete Walk-Jump Sampling. 6 2023. URL http://arxiv.org/abs/2306.12360.

[15] Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. Generative Language Modeling for Antibody Design. *bioRxiv*, 2022. doi: 10.1101/2021.12.13.472419. URL https://doi.org/10.1101/2021.12.13.472419.

[16] J Dauparas, I Anishchenko, N Bennett, H Bai, R J Ragotte, L F Milles, B I M Wicky, A Courbet, R J De Haas, N Bethel, P J Y Leung, T F Huddy, S Pellock, D Tischer, F Chan, B Koepnick, H Nguyen, A Kang, B Sankaran, A K Bera, N P King, and D Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378:49–56, 2022. URL https://www.science.org.

[17] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022. doi: 10.1101/2022.04.10.487779. URL https://doi.org/10.1101/2022.04.10.487779.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury Google, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf Xamla, Edward Yang, Zach Devito, Martin Raison Nabla, Alykhan Tejani, Sasank Chilamkurthy, Qure Ai, Benoit Steiner, Lu Fang Facebook, Junjie Bai Facebook, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, 2019.

[19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020. URL https://github.com/huggingface/.

[20] 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, 1 2022. ISSN 1469896X. doi: 10.1002/pro.4205.

[21] Baris E. Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H. Wu. UniRef: Comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–1288, 5 2007. ISSN 13674803. doi: 10.1093/bioinformatics/btm098.

[22] Jeffrey A. Ruffolo, Jeremias Sulam, and Jeffrey J. Gray. Antibody structure prediction using interpretable deep learning. *Patterns*, 3(2), 2 2022. ISSN 26663899. doi: 10.1016/j.patter.2021.100406.

[23] Dora Pinto, Young Jun Park, Martina Beltramello, Alexandra C. Walls, M. Alejandra Tortorici, Siro Bianchi, Stefano Jaconi, Katja Culap, Fabrizia Zatta, Anna De Marco, Alessia Peter, Barbara Guarino, Roberto Spreafico, Elisabetta Cameroni, James Brett Case, Rita E. Chen, Colin Havenar-Daughton, Gyorgy Snell, Amalio Telenti, Herbert W. Virgin, Antonio Lanzavecchia, Michael S. Diamond, Katja Fink, David Veesler, and Davide Corti. Cross-neutralization of SARS-CoV-2 by a human monoclonal SARS-CoV antibody. *Nature*, 583(7815):290–295, 7 2020. ISSN 14764687. doi: 10.1038/s41586-020-2349-y.

[24] Tongqing Zhou, Lingshu Wang, John Misasi, Amarendra Pegu, Yi Zhang, Darcy R. Harris, Adam S. Olia, Chloe Adrienna Talana, Eun Sung Yang, Man Chen, Misook Choe, Wei Shi, I. Ting Teng, Adrian Creanga, Claudia Jenkins, Kwanyee Leung, Tracy Liu, Erik Stephane D. Stancofski, Tyler Stephens, Baoshan Zhang, Yaroslav Tsybovsky, Barney S. Graham, John R. Mascola, Nancy J. Sullivan, and Peter D. Kwong. Structural basis for potent antibody neutralization of SARS-CoV-2 variants including B.1.1.529. *Science*, 376(6591), 4 2022. ISSN 10959203. doi: 10.1126/science.abn8897.

[25] Jeremy M Berg, John L Tymoczko, and Lubert Stryer. *Biochemistry (Loose-Leaf)*. Macmillan, 2007.

[26] James Dunbar and Charlotte M. Deane. ANARCI: Antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 1 2016. ISSN 14602059. doi: 10.1093/bioinformatics/btv552.

[27] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. Mcgettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, 11 2007. ISSN 13674803. doi: 10.1093/bioinformatics/btm404.

[28] Peter J.A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J.L. De Hoon. Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 6 2009. ISSN 13674803. doi: 10.1093/bioinformatics/btp163.

[29] Stephen F Altschup, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic Local Alignment Search Tool. *J. Mol. Biol*, 215:403–410, 1990.

[30] Ammar Tareen and Justin B Kinney. Logomaker: Beautiful Sequence Logos in Python. *biorxiv*, 2019. doi: 10.1101/635029. URL `https://doi.org/10.1101/635029`.

[31] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8. November 2015.

[32] Justas Dauparas, Haobo Wang, Avi Swartz, Peter Koo, Mor Nitzan, and Sergey Ovchinnikov. Unified framework for modeling multivariate distributions in biological sequences. *arXiv*, 2019. URL `http://arxiv.org/abs/1906.02598`.

[33] Kevin K Yang, Niccolò Zanichelli Eleutherai, and Hugh Yeh. Masked inverse folding with sequence transfer for protein representation learning. *bioRxiv*, 2022. doi: 10.1101/2022.05.25.493516. URL `https://www.biorxiv.org/content/early/2022/05/28/2022.05.25.493516`.

[34] Patrick Koenig, Chingwei V. Lee, Benjamin T. Walters, Vasantharajan Janakiraman, Jeremy Stinson, Thomas W. Patapoff, and Germaine Fuh. Mutational landscape of antibody variable domains reveals a switch modulating the interdomain conformational dynamics and antigen binding. *Proceedings of the National Academy of Sciences of the United States of America*, 114 (4):E486–E495, 2017. ISSN 10916490. doi: 10.1073/pnas.1613231114.

[35] Shira Warszawski, Aliza Borenstein Katz, Rosalie Lipsh, Lev Khmelnitsky, Gili Ben Nissan, Gabriel Javitt, Orly Dym, Tamar Unger, Orli Knop, Shira Albeck, Ron Diskin, Deborah Fass, Michal Sharon, and Sarel J. Fleishman. Optimizing antibody affinity and stability by the automated design of the variable light-heavy chain interfaces. *PLoS Computational Biology*, 15 (8):1–24, 2019. ISSN 15537358. doi: 10.1371/journal.pcbi.1007207.

[36] Brian L. Hie, Duo Xu, Varun R. Shanker, Theodora U.J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models and sequence information alone. *bioRxiv*, page 2022.04.10.487811, 2022. URL https://www.biorxiv.org/content/10.1101/2022.04.10.487811v1%0Ahttps://www.biorxiv.org/content/10.1101/2022.04.10.487811v1.abstract.

Contents

A Appendix

A.1 Method

A.1.1 Dataset

A visualization of dataset splitting strategy is given by Figure A.1.

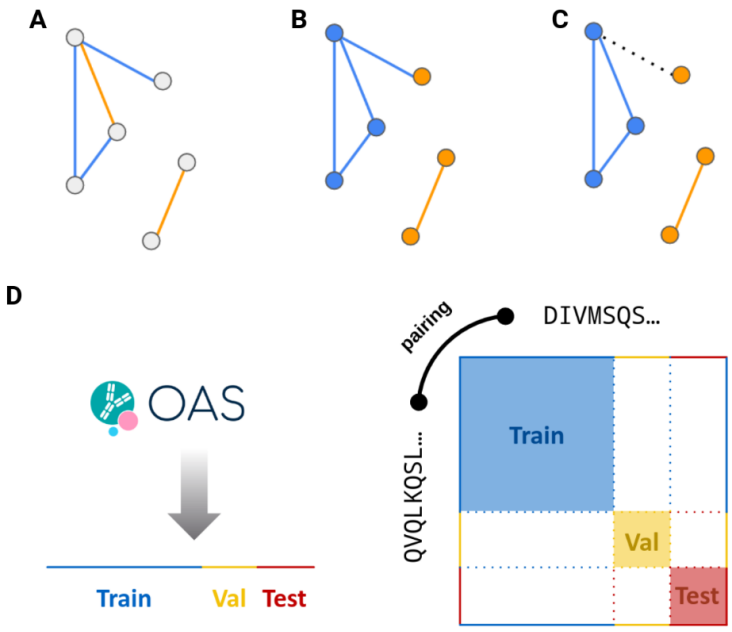


Figure A.1: Splitting for protein-protein interaction dataset. Each sequence and pairing is represented by a node and an edge respectively, colorized by train (blue) and test (orange) partitions. (A) Interaction split. Nodes are not partitioned and are therefore colorless. (B) Inclusive node split. (C) Exclusive node split. Edges between train and test nodes are dropped (dotted line). (D) Exclusive node splitting in detail. All non-redundant sequences in paired OAS database are first split into train, validation, and test partitions. Only pairings within each partition are included in the final dataset, i.e. all cross-pairings are dropped.

A.2 Pairing Perplexity Reflects Preferences in Chain Pairing

To demonstrate that our model understands the context of antibody pairing, we evaluate the model based on the perplexity of the sequence pairs. Using the human LM T5 in English-to-German translation as an analogy, feeding an English sentence to the encoder and its German counterpart to the decoder should in general yield a lower perplexity than feeding both encoder and decoder with English sentences. The idea is to probe the model's capability to understand that a German sentence should be generated from an English input in a generative model, instead of assessing the model in a traditional sentence-pair classification task.

Without publicly available antibody mispairing dataset, we test our model on two simple mispairing scenarios, i.e. chain-type mispairing and species mispairing. For chain-type mispairing, we synthesize *correct* heavy-light pairing and *mispaired* heavy-heavy/light-light pairing for each translation in test set, with the assumption that only heavy-light-chain pairings are permitted. A similar approach is used for species mispairing by assuming cross-species chain pairing is impermissible. Note that, given the promiscuous nature of antibody chain pairing, a heavy chain sequence can pair with multiple light chain chain sequences. Therefore, randomly paired heavy and light chain can still be a valid pairing and cannot serve as a negative control in comparison to observed pairing by contrasting their respective perplexity.

We propose two classification tasks (Figure A.2) to assess our model's perplexity. The first task considers two input sequences sharing the same target sequence and only one pairing is correct.

Out of the two pairings, we assign the pairing with lower perplexity as *correct* and the other one as *mispaired*. Based on this assignment, we identify above 90% of the *correct* pairings from chain-type mispairing and close to 80% from species mispairing. The baseline of random assignment results in 50% accuracy. No classification model is trained.

In our second task, we consider a dataset by mixing and shuffling the *correct* and *mispaired* samples from the first task and classify whether the pairing is *correct* given two antibody sequences alone. Informed only by our language model's perplexity, a logistic regression significantly outperforms the baseline of random assignment. The classifier is trained on the average perplexity of forward- and back-translations on validation set. All performance metrics are evaluated on test set. The weaker classification performance might be attributed to the loss of pairing preferences between gene loci and families in the creation of mispairing dataset (Subsection A.2).

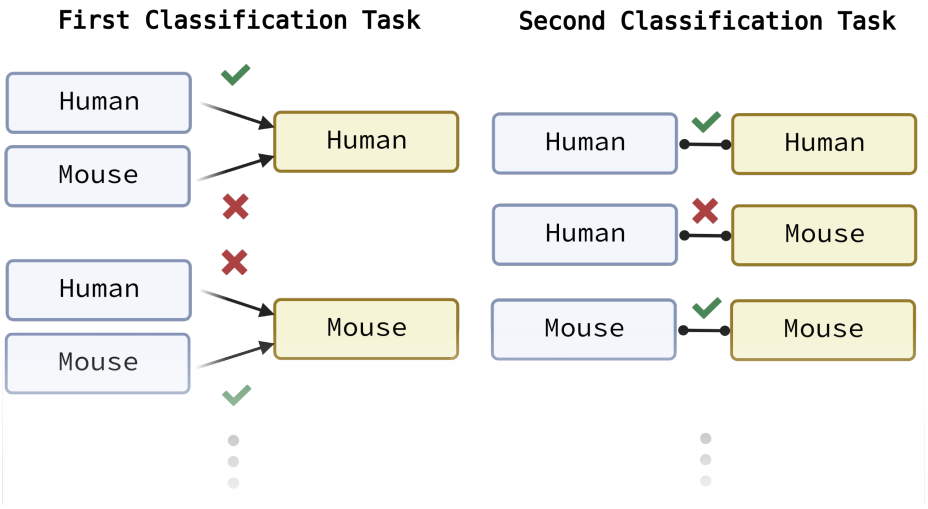


Figure A.2: Schematics of two classification tasks considered for species mispairing. (Left) In the first classification task, the aim is to identify the *correct* and *mispaired* sequences sharing the same target. (Right) In the second classification task, the aim is to predict the likelihood of the pairing as a bidirectional translation. The tasks for chain-type mispairing are similar. No chain type nor species annotation is used in our prediction.

First Classification Task

Mispairing type	Target chain	Accuracy
Chain type	Light	0.92
	Heavy	0.91
Species	Light	0.80
	Heavy	0.79

Second Classification Task

Mispairing type	Accuracy	AUROC
Chain type	0.54	0.70
Species	0.57	0.60

Table A.1: Performance on first and second classification task on model perplexity alone. (Up) In the first classification task, mispairing assignment is based on the rank of perplexity without any parameterizable model. (Bottom) In the second classification task, logistic regression is trained on the bidirectional average of translation perplexity in validation set, and evaluated on test set. Random assignment results in an accuracy of 0.5 in the first task, and an additional AUROC of 0.5 in the second task.

To further elaborate on the methodology, we generate synthetic mispairings to test our model's capability of learning chain pairing. The generation protocol for chain-type mispairing is as follows (algorithm 1). The generation protocol for species mispairing is similar (algorithm 2).

Algorithm 1 Chain-type mispairing dataset generation

Inputs: paired test dataset D
Outputs: chain-type mispairing dataset D'
initialize H, L and D' as $\emptyset$
for (u, v) in D **do**
    for s in (u, v) **do**
        if chaintype(s) = heavy **then**
            H.add(s)
        else if chaintype(s) = light **then**
            L.add(s)
        end if
    end for
end for
for (u, v) in D **do**
    if chaintype(u) = chaintype(v) **then**
        for s in (u, v) **do**
            if chaintype(s) = heavy **then**
                $s' \leftarrow random\ element\ in\ L$
            else if chaintype(s) = light **then**
                $s' \leftarrow random\ element\ in\ H$
            end if
            D'.add((s, s'))
        end for
    end if
end for
return D'

Algorithm 2 Species mispairing dataset generation

Inputs: paired test dataset D
Outputs: species mispairing dataset D'
initialize H, M and D' as $\emptyset$
for (u, v) in D **do**
    for s in (u, v) **do**
        if species(s) = human **then**
            H.add(s)
        else if species(s) = mouse **then**
            M.add(s)
        end if
    end for
end for
for (u, v) in D **do**
    if species(u) = species(v) **then**
        for s in (u, v) **do**
            if species(s) = human **then**
                $s' \leftarrow random\ element\ in\ M$
            else if species(s) = mouse **then**
                $s' \leftarrow random\ element\ in\ H$
            end if
            D'.add((s, s'))
        end for
    end if
end for
return D'

We have considered two possible schemes for preparing correct pairings (Figure A.3), i.e. single-generation and double-generation. In single-generation, we keep the observed pairing from test set as the correct pairing. While it ensures that the correct pairing is experimentally validated, the comparison between an observed correct pairing and a synthetic mispairing creates a bias in perplexity.

As such, we introduce double-generation where both pairings are generated and label the synthetically *correct* pairing in italic. Despite the lack of direct experiment validation, the comparison between correct and mispaired pairings is unbiased, is more challenging than single-generation, and provides some insights into whether our model learns antibody chain pairing. As indicated in Table A.2 and A.3, the conclusion remains the same when switched from single-generation to double-generation.

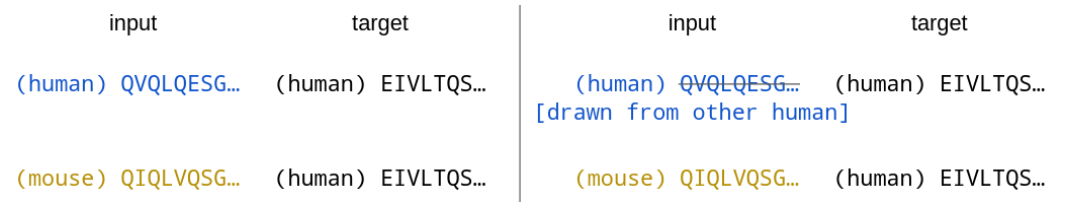


Figure A.3: Schematics of preparation of correct and mispaired sequences in species mispairing. The input sequence for correct pairing is in blue and that for mispairing is in yellow. (Left) Single-generation scheme: comparison between observed correct pairing and synthetic mispairing. (Right) Double-generation scheme: comparison between synthetic *correct* pairing and synthetic mispairing.

Mispairing type	Target chain	Accuracy
Chain type	Light	0.99
	Heavy	0.96
Species	Light	0.97
	Heavy	0.96

Table A.2: First classification task assignment accuracy by the perplexity rank between correct and *mispaired* antibody sequences in single-generation scheme.

Mispairing type	Accuracy	AUROC
Chain-type	0.54	0.72
Species	0.60	0.70

Table A.3: Second classification task performance in single-generation scheme

A.3 Conditional Generation Recovers Pairing Sequences

In order to evaluate our model's sequence-to-sequence generative performance, we test whether our model can recover the observed pairing in test set. Figure A.5 illustrates the recovery rate at progressively fine levels of resolution on human antibodies. A target sequence is considered to be recovered if the generated sequence shares the same chain type, gene loci, V gene family, or the combination of V and J gene families. For chain types, our model always generates heavy chains from light chain inputs, and likewise for light chain generation. For gene loci on light chain, λ and κ loci are recovered at 48% and 56% of the time. As we approach finer resolutions, the recovery rate drops in V families and their combination with J families. This is consistent with the observation that antibody chain pairing is often degenerate. For instance, the heavy chain sequences from IGHV1 gene family are observed to pair with multiple families in both λ and κ loci (Figure A.11). This sets an upper bound on the recovery rate in antibody heavy and light chain pairing. A similar analysis has

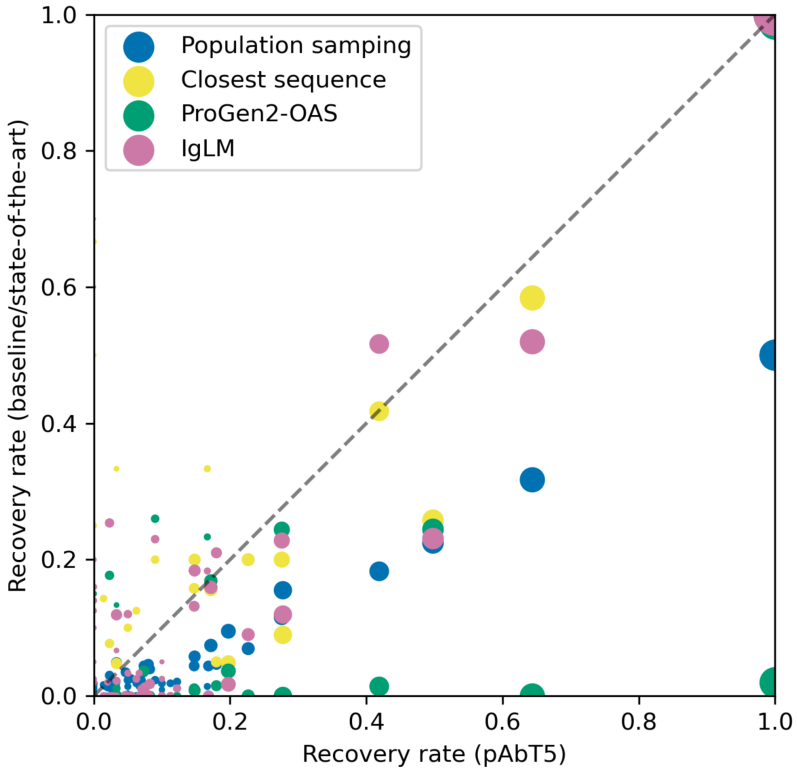


Figure A.4: Percentage of generated sequences sharing the same chain type, gene loci, V gene families, and V-J gene families with target. x-axis is the recovery rate of pAbT5, and y-axis is the recovery rate of ProGen2-OAS [13], IgLM [15], picking a sequence randomly from the population, and picking the pairing partner of the closest sequences from the validation set. Each scatter point represents recovery at a resolution, and size of the scatter point is proportional to its respective population size. The full table is available in Supplementary Materials.

also been performed on the recovery of species (Figure A.9) and the exact figures of recovery rate can depend on the generative parameters, which are listed in Subsection A.3.

We use ANARCI [26] for species, chain type, and gene family classification. Although OAS dataset indicates humans, mice, and rats as the source organisms, ANARCI identifies only the former two. For consistent comparison in both observed and generated antibody pairs, we opt for the definition in ANARCI in all evaluations, including t-SNE, mispairing, and generation assessment. We only report V and J families in heavy and light chains as D families are not supported by ANARCI. In all species-specific analyses, pairings are included only when ANARCI identifies both heavy and light chains from the same species.

We denote the encoder sequence as the input of the translation and decoder sequence as the target of the translation. We denote the encoder hidden state of the paired antibody in the translation order of input-to-target as the sequence embedding of the input sequence, or simply sequence embedding. For t-SNE visualization, we take the mean of the encoder hidden state over residues at the final layer.

In the generative process, sequences are generated at a temperature of 1, top p of 0.9 with 10 returned sequences, determined from a grid search of temperature and top p. Experiment on beam search results in low diversity and regions of repetitive motifs. All co-occurrences of gene families are collected from test set. For ProGen2-OAS [13], we use the default generative parameters and do not provide the first few tokens to avoid hinting at the chain type and gene loci. We use default generative parameters in IgLM [15].

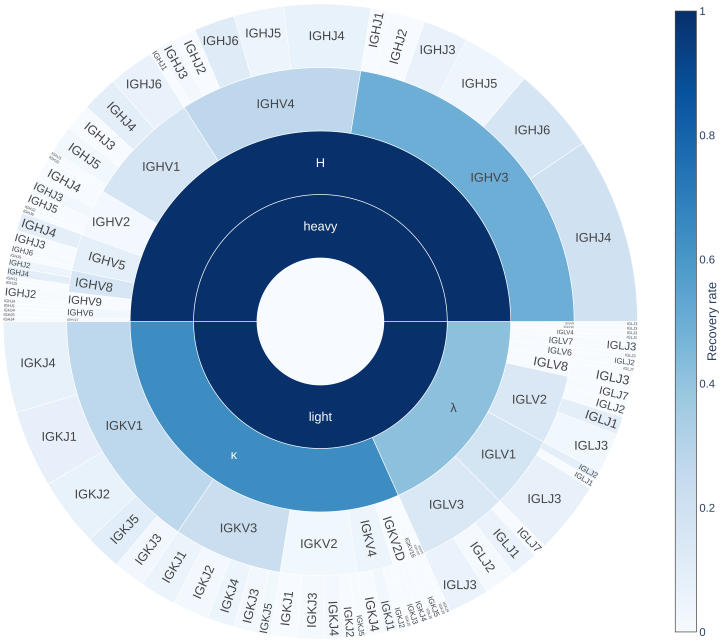


Figure A.5: Recovery rate of target chain type, gene loci, and gene families in sequence generation. Performance is represented in a hierarchical order, where parent classes are centered while children categories are on the periphery. On each rim, the arc lengths of categories are proportional to their populations in test set. Dark blue represents perfect recovery whereas white color implies low recovery rate.

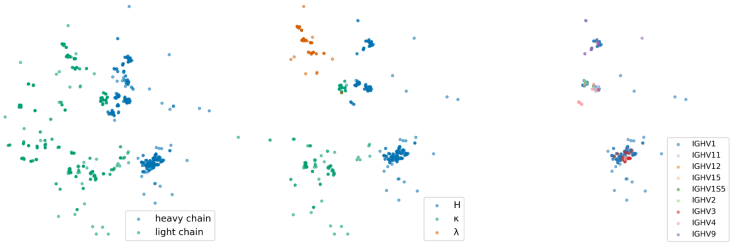


Figure A.6: t-SNE plot of encoder hidden states of test set sequences in progressively fine categories (chain types, human gene loci, and human IGHV gene families).

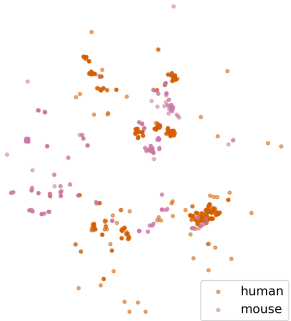


Figure A.7: t-SNE plot of sequence embeddings colorized by ANARCI annotated species

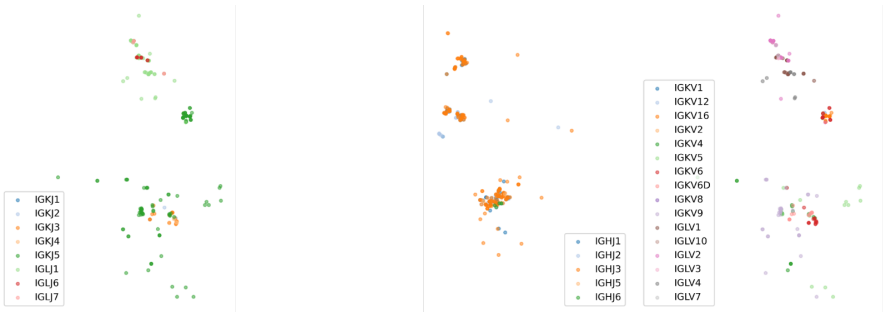


Figure A.8: t-SNE plot of sequence embeddings colorized by ANARCI annotated gene families. (Left) Light chain J gene. (Middle) Heavy chain J gene. (Right) Light chain V gene.

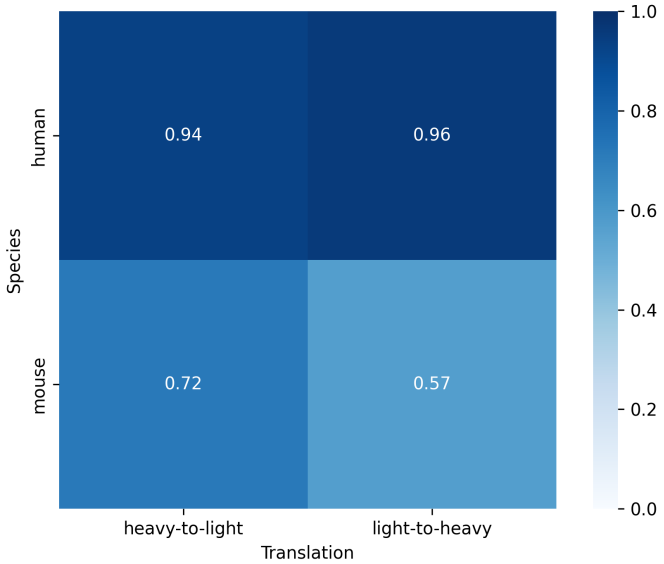


Figure A.9: Recovery rate on species by original species and translation direction.

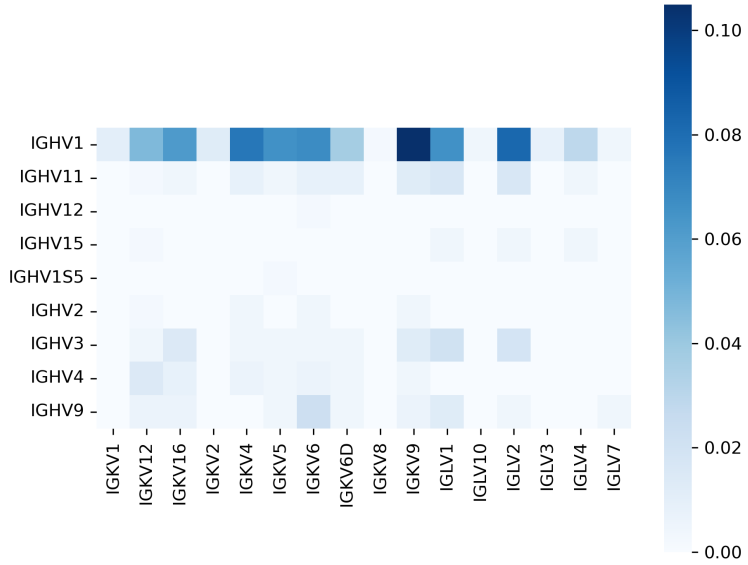


Figure A.10: Co-occurrence of V families in heavy and light chains colorized by relative frequency. Frequency is normalized by the total number of observed co-occurrence.

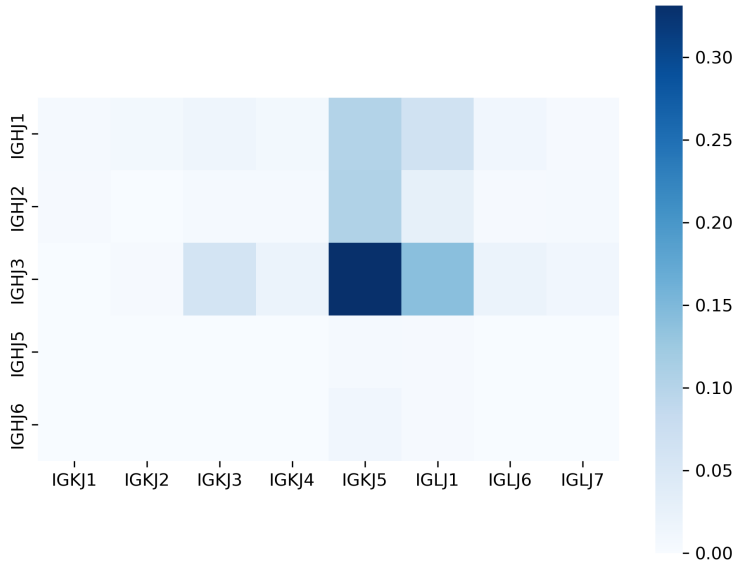


Figure A.11: Co-occurrence of J families in heavy and light chains colorized by relative frequency. Frequency is normalized by the total number of observed co-occurrence.

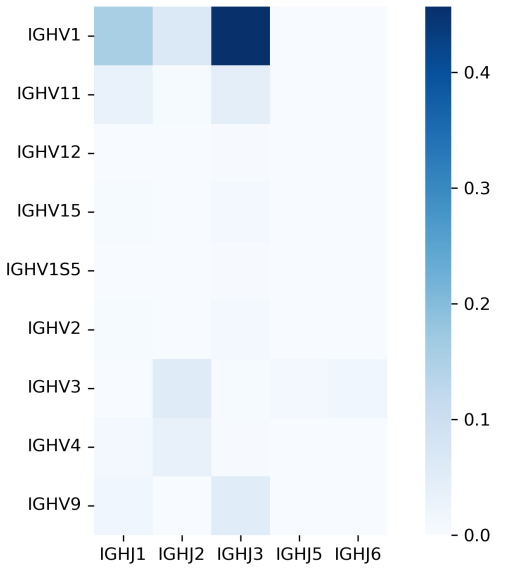


Figure A.12: Co-occurrence of V and J families in heavy chain colorized by relative frequency. Frequency is normalized by the total number of observed co-occurrence.

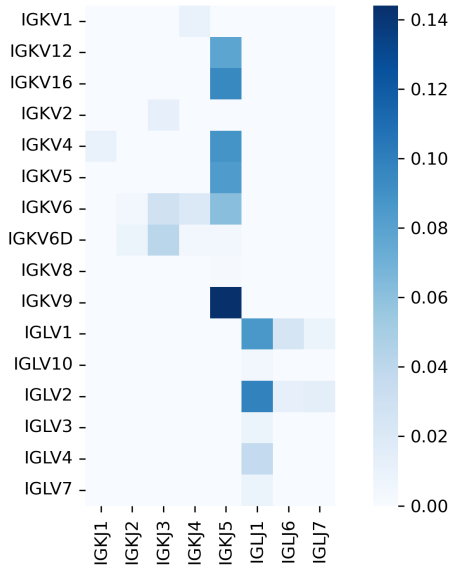


Figure A.13: Co-occurrence of V and J families in light chain colorized by relative frequency. Frequency is normalized by the total number of observed co-occurrence.

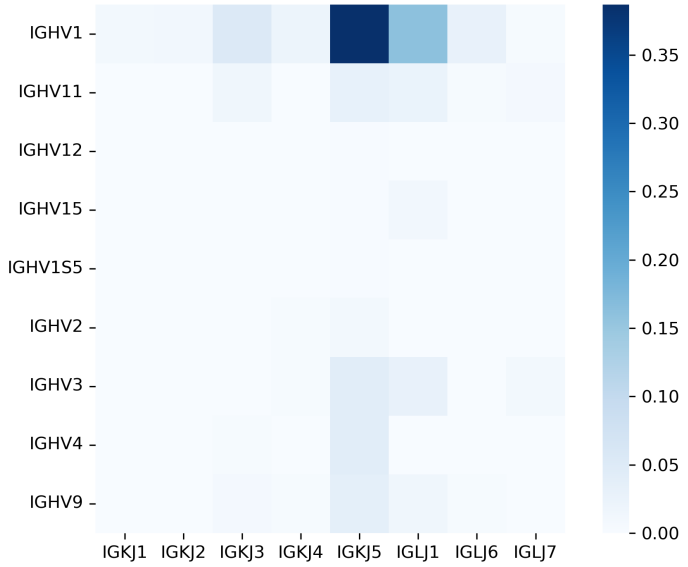


Figure A.14: Co-occurrence of V families in heavy chain and J families in light chain colorized by relative frequency. Frequency is normalized by the total number of observed co-occurrence.

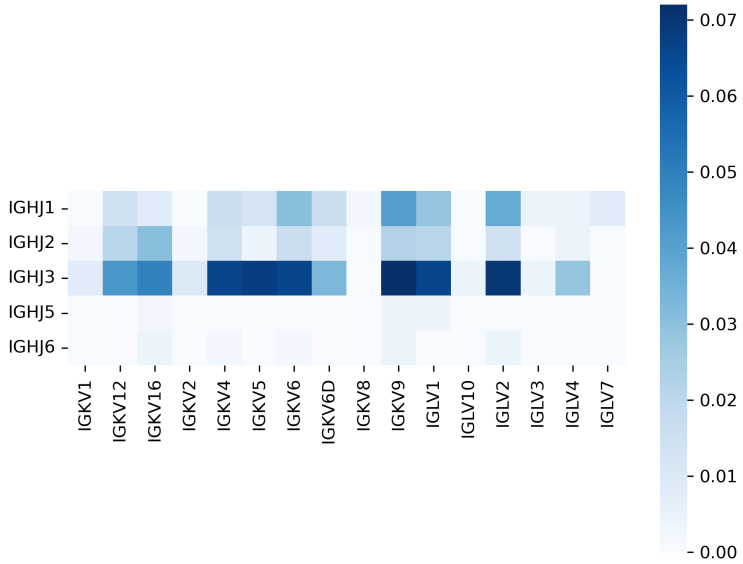


Figure A.15: Co-occurrence of J families in heavy chain and J families in light chain colorized by relative frequency. Frequency is normalized by the total number of observed co-occurrence.

A.4 Sequence Generation Aligns with Conserved and Variable Domains in Antibodies

We use clustalw [27] in Biopython [28] with default parameters to generate alignment profiles. Conservation analysis is generated by psiblast [29] in Biopython onto UniRef90 database [21]. To compare model confidence and sequence conservation, we apply softmax to PSSM and compare with the probability in next-word prediction. We use Logomaker [30] for visualization of sequence and alignment profiles. CDR and framework regions are defined in aho antibody renumbering scheme. CDRs of light chains are from residue ID 32 to 42, 57 to 76, and 109 to 138 for CDR L1, L2, and L3 respectively. CDRs of heavy chains are located from residue ID 24 to 42, 58 to 72, and 107 to 138.

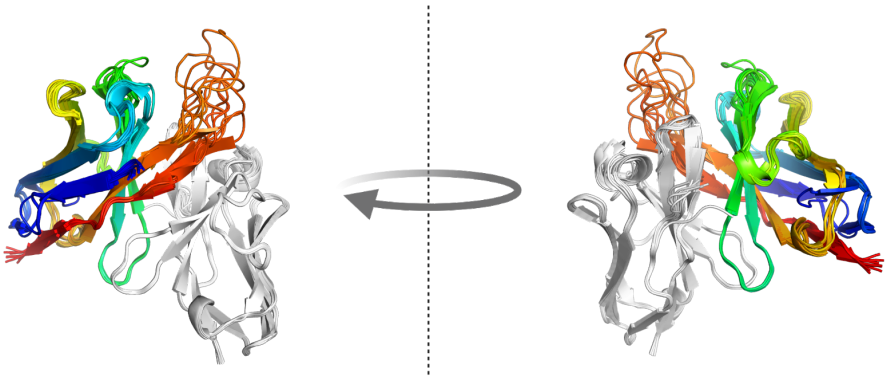


Figure A.16: Structural models of example variable regions (Fv) with eight generated heavy chains given one input light chain. The generated heavy chains are colored in rainbow and the light chains are white.

We overlay entropy and cross-attention per query residue onto antibody structures in PyMOL [31]. Structural models are generated from DeepAb [22], and in the case with available crystal structures, we align the models to the crystal chains to standardize numbering and fill in missing residues. We cap the values of average entropy and cross-attention per query residue in structural overlay and normalize heavy and light chains together for visualization purposes. Detailed visualization of capped and uncapped figures are also available (Figure A.23, A.24, A.25, and A.26).

Region	Light	Heavy
FR1	0.57±0.18	0.63±0.21
CDR1	0.36±0.26	0.41±0.22
FR2	0.77±0.13	0.76±0.14
CDR2	0.38±0.21	0.41±0.19
FR3	0.71±0.12	0.63±0.18
CDR3	0.31±0.20	0.22±0.14
FR4	0.76±0.18	0.90±0.09
whole sequence	0.60±0.13	0.59±0.14

Table A.4: Sequence identities between generated and target sequences in test set by regions and target chain type.

	Heavy chain target	Light chain target
Human	0.61±0.14	0.60±0.14
Mouse	0.56±0.10	0.62±0.10

Table A.6: Sequence identities between generated and target sequences in test set by species and target chain type

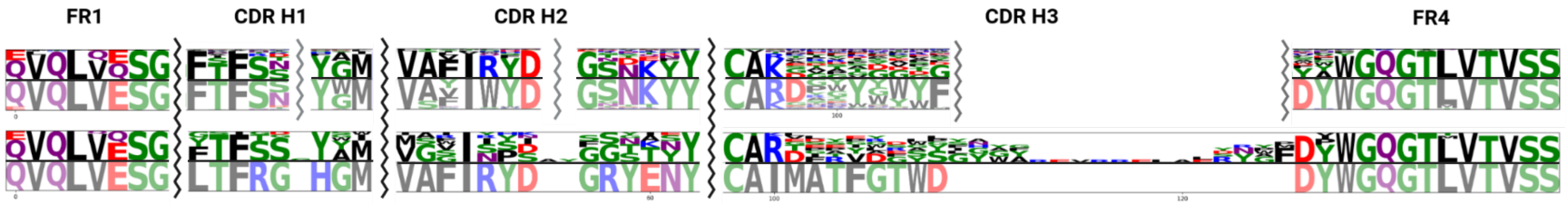


Figure A.17: Comparison between observed and modeled alignment profiles on heavy chain in framework regions (FRs) CDR loops. (First row) Next-word probability in teacher-forcing. (Second row) Sequence conservation from position-specific scoring matrix. (Third row) Global alignment of generated sequences to (fourth row) the observed sequence. In general, generated sequences are more variable than next-word probability due to the cascade effect in iterative sampling, and might have different gene locus and/or families from the target sequence. The full-length alignment profiles of heavy and light chains together with four other output examples randomly from test set are available in Figure A.19, A.20 and A.21.

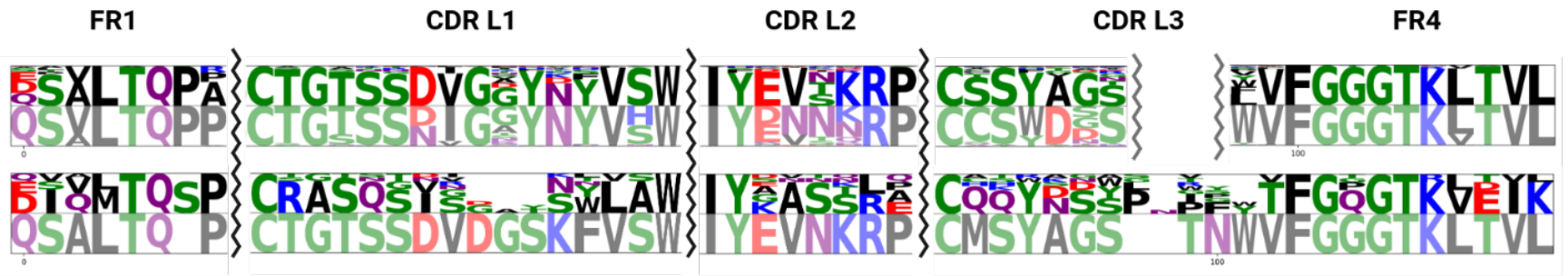
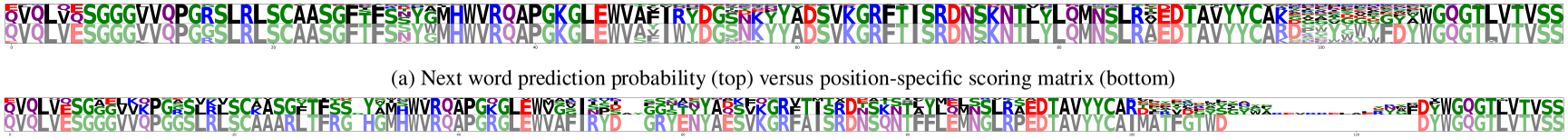


Figure A.18: Comparison between observed and modeled alignment profiles on heavy chain in framework regions (FRs) CDR loops. (First row) Next word prediction probability. (Second row) Sequence conservation from position-specific scoring matrix. (Third row) Global alignment of generated sequences to (fourth row) the observed sequence. The heavy chain in Figure 1 and the light chain here originate from the same observed antibody chain pair.



(a) Next word prediction probability (top) versus position-specific scoring matrix (bottom)

(b) Generated sequences (top) versus observed sequence (bottom)

Figure A.19: Full-length alignment profile of heavy chain between model predictions, conservation profile and observed sequence in Figure 1.

(a) Next word prediction probability (top) versus position-specific scoring matrix (bottom)

(b) Generated sequences (top) versus observed sequence (bottom)

Figure A.20: Full-length alignment profile of light chain between model predictions, conservation profile and observed sequence in Figure A.18.

(a) Next word prediction probability (top) versus position-specific scoring matrix (bottom)

(b) Generated sequences (top) versus observed sequence (bottom)

(c) Next word prediction probability (top) versus position-specific scoring matrix (bottom)

(d) Generated sequences (top) versus observed sequence (bottom)

(e) Next word prediction probability (top) versus position-specific scoring matrix (bottom)

(f) Generated sequences (top) versus observed sequence (bottom)

(g) Next word prediction probability (top) versus position-specific scoring matrix (bottom)

(h) Generated sequences (top) versus observed sequence (bottom)

Figure A.21: Four other examples of full-length alignment profile of light chain between model predictions, conservation profile and observed sequence. Examples are randomly drawn from all test set translations.

Region	Light		Heavy	
	Observed	Generated	Observed	Generated
FR1	22.75±0.43	22.74±0.44	28.91±1.45	28.99±0.06
FR2	15.00±0.00	15.00±0.00	14.00±0.00	14.00±0.00
FR3	32.02±0.20	32.00±0.04	32.00±0.00	32.00±0.00
FR4	9.97±0.22	10.00±0.03	11.00±0.00	10.96±0.33
CDR1	12.50±2.16	12.54±2.14	6.32±0.75	6.22±0.62
CDR2	7.03±0.38	7.02±0.25	16.80±0.77	16.82±0.66
CDR3	9.24±0.96	9.44±1.01	11.47±4.00	12.29±4.16
whole sequence	108.51±2.38	108.74±2.29	120.45±4.57	121.28±4.18

Table A.5: Sequence length of observed and generated sequences in test set by regions and chain type.

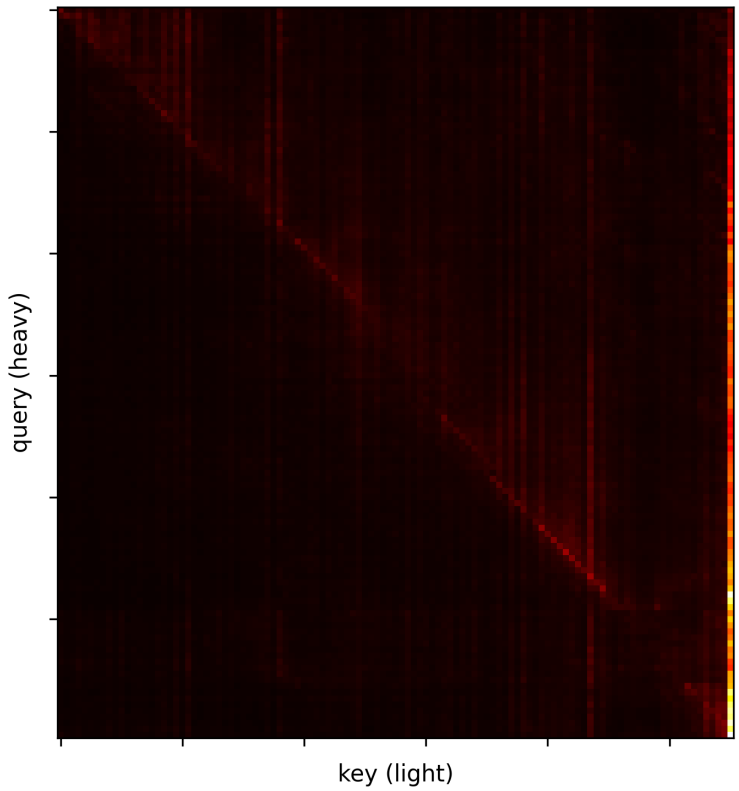


Figure A.22: Cross-attention map between target heavy chain and input light chain in Figure 1 averaged throughout heads and layers. Hypervariable regions generally receive less attention from queries consistently throughout all paired antibodies in the test set.

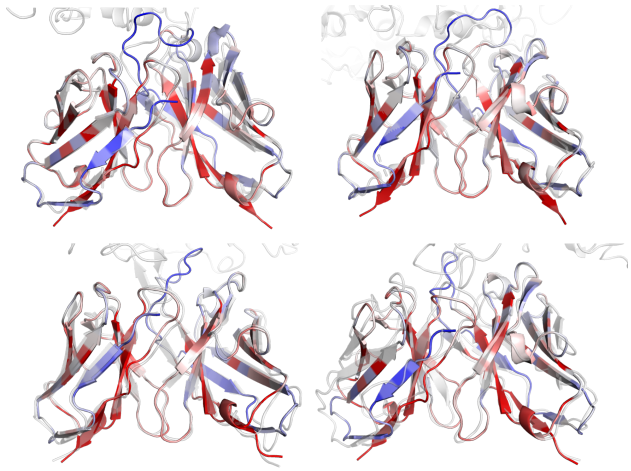


Figure A.23: Structural overlay of capped average cross-attention from pairing partner onto each residue of SARS-Cov2-binding antibodies. Red color indicates regions highly attended while blue is weakly attended areas. (Upper right) PDB 6WPT. (Lower Left) PDB 7TB8 chain D and E. (Lower Right) PDB 7TB8 chain H and I. Consistently for all PDB structures, the CDR loops receive the least attention. This is consistent with the random nature of CDR loop sequences.

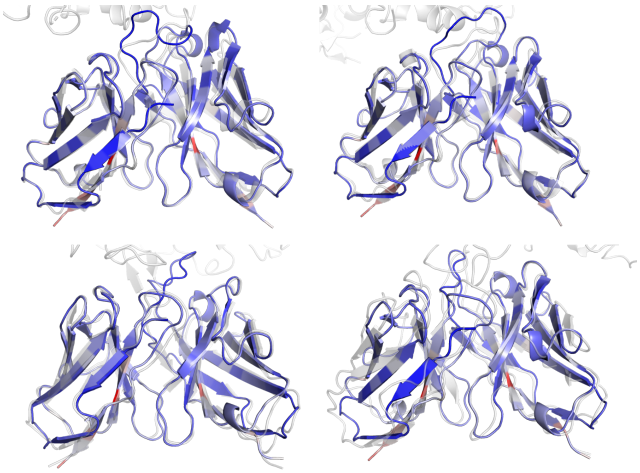


Figure A.24: Structural overlay of uncapped average cross-attention from pairing partner onto each residue of SARS-Cov2-binding antibodies. (Upper right) PDB 6WPT. (Lower Left) PDB 7TB8 chain D and E. (Lower Right) PDB 7TB8 chain H and I.

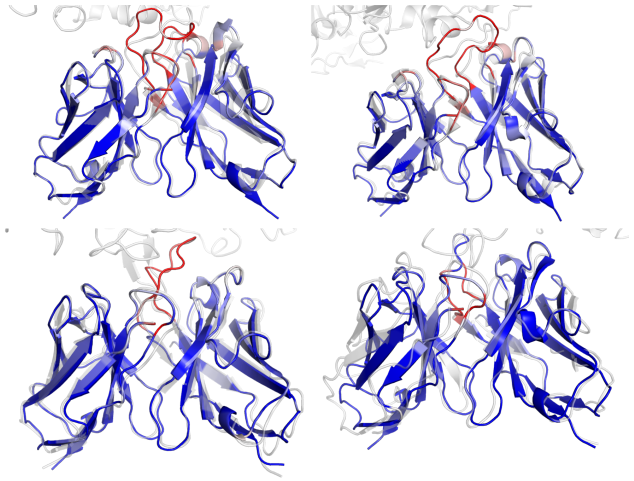


Figure A.25: Structural overlay of capped next word prediction entropy of SARS-Cov2-binding antibodies. (Upper left) PDB 6WPS. (Upper right) PDB 6WPT. (Lower Left) PDB 7TB8 chain D and E. (Lower Right) PDB 7TB8 chain H and I.

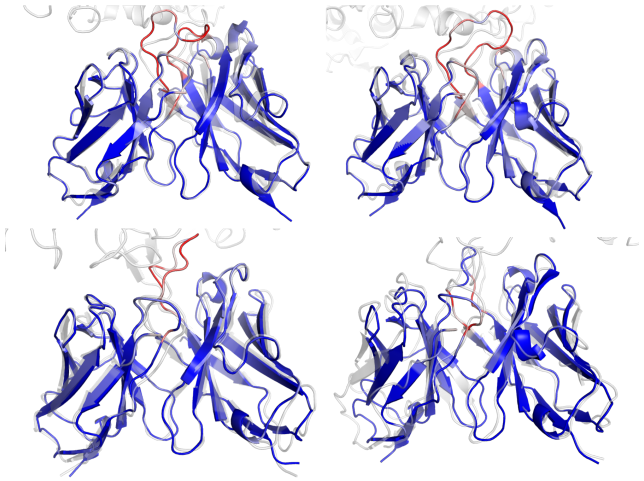


Figure A.26: Structural overlay of uncapped next word prediction entropy of SARS-Cov2-binding antibodies. (Upper left) PDB 6WPS. (Upper right) PDB 6WPT. (Lower Left) PDB 7TB8 chain D and E. (Lower Right) PDB 7TB8 chain H and I.

A.5 Zero-shot Prediction from Paired Antibody Perplexity

The emergence of protein function prediction from sequences alone can be traced back to conservation analysis. The idea is that residues detrimental to the function(s) of the protein should be conserved while other positions have more freedom to vary. Encoder-only protein LMs were shown to generalize Pott's model [32], and outperform positional-specific scoring matrix (PSSM) with zero-shot prediction [11]. Similarly, the perplexity of decoder-only models is found to correlate with unseen experiment measurements [13], while the same log-likelihood analysis can also be replicated on conditional sequence generation in inverse folding [17, 33]. Zero-shot and few-shot predictions from language pretraining are not unique to protein LMs but arise generally from large-scale language modeling.

Benchmarked on antibody functional datasets, we show that our model has competitive results with the current state-of-the-art protein LMs. We benchmark our model on 13 antibody functional datasets on either stability, binding affinity or expression measurements [34–36] in Figure A.27. Our encoder-decoder model achieves a similar performance as ProGen2 and is better than ProGen2-

OAS, which is finetuned on the unpaired OAS dataset. The major architectural difference is that ProGen2 is a decoder-only model which requires joining heavy and light chain sequences with a GS linker, whereas our encoder-decoder model computes the average perplexity of forward- and back-translations. Nonetheless, ProGen2 and ProGen2-OAS have fewer parameters than our model, making model comparison difficult. In addition, we have also included pseudo-perplexity from encoder-only models (ESM) [1, 11] to highlight the difference in architecture.

To further investigate the impact of each component in our model, we perform an ablation study on the need for an encoder-decoder architecture, bidirectional translations in evaluation, and pretraining. For any comparison with statistical significance (p-value < 0.05), our encoder-decoder model always outperforms ablations (Figure A.28).

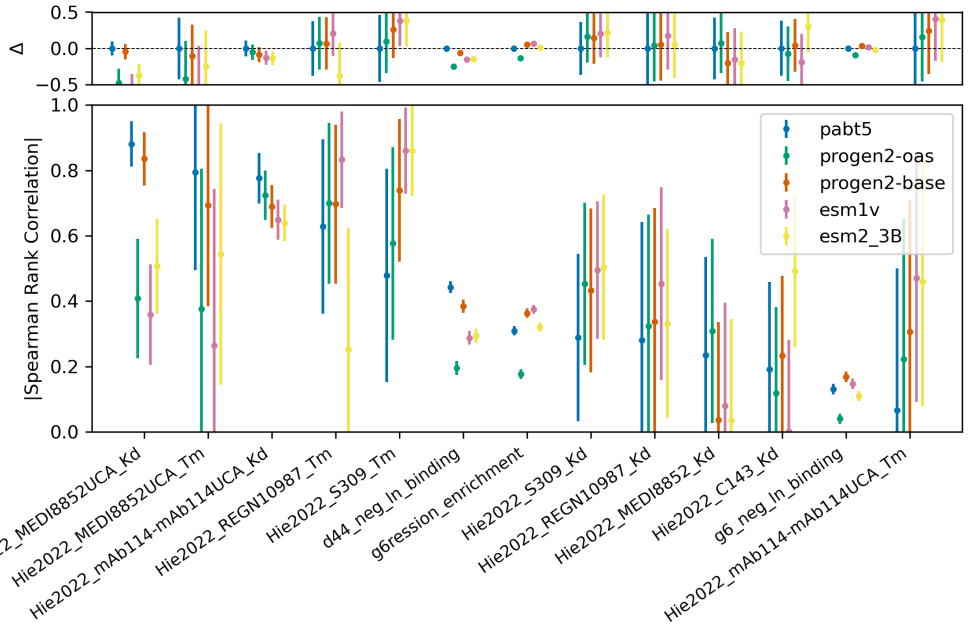


Figure A.27: Zero-shot prediction performance on antibody measurements of our model and state-of-the-art. x-axis represents the antibody functional datasets. (Top) The difference in absolute spearman rank correlation (SRC) between our model and state-of-the-art. (Bottom) Absolute SRC between model (pseudo-)perplexity and measurements. Error bars are estimated in standard deviation with 1000 bootstrap samples.

We evaluate the perplexity from the benchmarked models and calculate the absolute value of spearman rank correlation (SRC) with the experimental measurements. By default, we define a symmetric paired perplexity by taking the average of that in forward- and back-translations for zero-shot prediction. Since ProGen2 is a decoder-only model, we join the heavy and light chains by a GS linker of *GGGGSGGGGSGGGGS* and parse the paired antibody as a single sequence. In the case of our decoder-only ablation, we train the model without an encoder but take the average of heavy and light chain perplexities. Our ablation on pretraining from ProtT5 shares the same hyperparameters in Section **??**. The mean and standard deviation of SRC are estimated by bootstrapping 1000 samples.

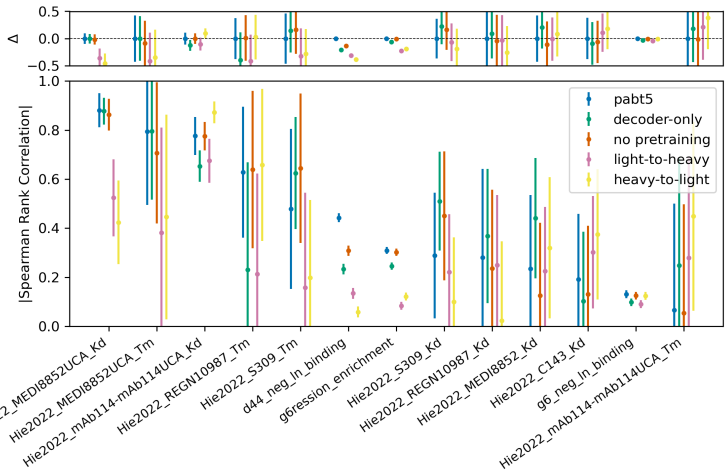


Figure A.28: Ablation study on zero-shot prediction on all datasets. x-axis represents datasets. (Top) The difference in absolute spearman rank correlation (SRC) between our model and ablation. (Bottom) Absolute SRC between model (pseudo-)perplexity and measurements. Error bars are estimated in standard deviation with 1000 bootstrap samples.

B Sequence Clustering

Contrary to using all non-redundant sequences in the dataset, one can cluster these sequences by an identity cutoff and include only the representative sequences of each cluster. This provides a few advantages. First, it reduces the dataset size and increases sparsity for efficient training. Second, it de-biases the database from heavily studied families. Third, it provides a better assessment of model generalizability by limiting the information shared between train and test sets.

This section investigates the impact of sequence clustering on paired OAS dataset and our model performance. We argue that for our specific case, including all non-redundant sequences helps the model in three ways. While sequence clustering affects the performance evaluation, the impact is minor and does not affect conclusions.

- Sequence clustering reduces the size of paired OAS dataset by at least 50%.
- Fine-grained resolution in a subspace of protein universe helps resolve all antibodies and their pairings, in particular for learning gene families.
- De-biasing might fail to reflect the preference(s) of antibody pairing.

B.1 Impact on Dataset Size

We use linclust from mmseqs2 to cluster representative sequences with –min-seq-id to specify identity cutoff, and -c 0.8 and –cov-mode 1, and otherwise the default parameters. We do not observe any signs of truncation at the N- and C-termini on paired OAS dataset.

As reported in Table B.1, the dataset reduces in size exponentially with the identity threshold in clustering. For each increment of 5%, the number of translations after clustering falls by about half. This impacts not only the training but also the statistical power of evaluation(s) given the size of the diminished test set.

From here, we denote exclusive node split in Section **??** on clustered sequences as cluster split. We decide to repeat the analyses on cluster split with an identity cutoff of 95% and compare with that from training on non-redundant sequences.

	non-redundant	95%	90%	85%
Training set	260062	127904	53814	22266
Validation set	846	356	188	74
Test set	802	346	178	78

Table B.1: Impact of identity threshold on dataset size in terms of number of translations

B.2 Impact on Results

B.2.1 Pairing Perplexity Reflects Preferences in Chain Pairing

In double-random scheme, training and evaluation on clustered sequences result in higher accuracy in the first classification task but weaker in the second classification task. In both tasks, mispairing identification informed by model perplexity alone still outperforms the baseline. Similar observation holds also in single-random scheme B.3 and B.4. Overall, the results are unaffected by sequence clustering.

First Classification Task

Mispairing type	Target chain	Accuracy
Chain type	Light	0.98
	Heavy	0.98
Species	Light	0.85
	Heavy	0.88

Second Classification Task

Mispairing type	Accuracy	AUROC
Chain type	0.55	0.65
Species	0.55	0.57

Table B.2: Performance on first and second classification task on model perplexity alone. (Left) In the first classification task, mispairing assignment is based on the rank of perplexity without any parameterizable model. (Right) In the second classification task, instead of unidirectional translation, logistic regression is trained on the bidirectional average of translation perplexity in validation set, and evaluated on test set. Random assignment results in an accuracy of 0.5 in the first class, and an additional AUROC of 0.5 in the second task.

Mispairing type	Target chain	Accuracy
Chain type	Light	0.92
	Heavy	1
Species	Light	0.99
	Heavy	0.98

Table B.3: First classification task assignment accuracy by the perplexity rank between correct and *mispaired* antibody sequences in single-generation scheme.

Mispairing type	Accuracy	AUROC
Chain-type	0.55	0.62
Species	0.56	0.62

Table B.4: Second classification task performance in single-generation scheme

B.2.2 Sequence Generation Aligns with Conserved and Variable Domains in Antibodies

Our model from cluster split still has high entropy and generates variable-length sequences at hypervariable domains. Results are largely unaffected by cluster split.

Figure B.1: Comparison between observed and modeled alignment profiles on heavy chain in framework regions (FRs) CDR loops. (First row) Next word prediction probability. (Second row) Sequence conservation from position-specific scoring matrix. (Third row) Global alignment of generated sequences to (fourth row) the observed sequence.

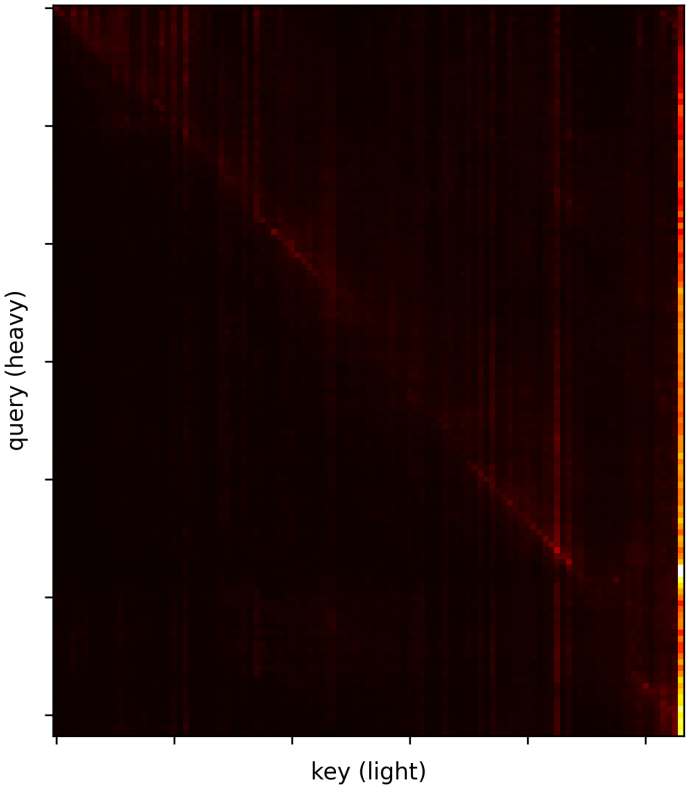


Figure B.2: Cross-attention map between target heavy chain and input light chain in Figure B.1 averaged throughout heads and layers. Hypervariable regions generally receive less attention from queries.

Region	Light	Heavy
FR1	0.59±0.18	0.60±0.21
CDR1	0.35±0.25	0.38±0.20
FR2	0.78±0.13	0.76±0.13
CDR2	0.39±0.22	0.37±0.15
FR3	0.69±0.12	0.58±0.15
CDR3	0.33±0.20	0.24±0.15
FR4	0.79±0.19	0.90±0.08
whole sequence	0.60±0.13	0.56±0.12

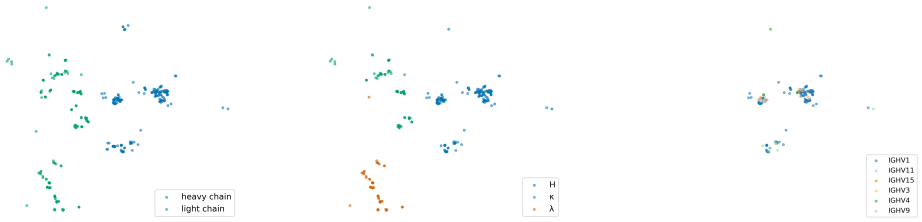
Table B.5: Sequence identities between generated and target sequences in test set by regions and target chain type.

Region	Light		Heavy	
	Observed	Generated	Observed	Generated
FR1	22.59±0.50	22.37±0.48	28.86±1.83	29.00±0.00
CDR1	12.74±2.18	12.87±1.41	6.26±0.67	6.03±0.25
FR2	15.00±0.00	15.00±0.00	14.00±0.00	14.00±0.00
CDR2	7.05±0.43	7.00±0.00	16.74±0.63	16.89±0.32
FR3	32.00±0.00	32.00±0.00	32.00±0.15	32.00±0.00
CDR3	9.63±1.06	9.95±0.81	12.32±3.93	16.51±3.97
FR4	9.94±0.36	10.00±0.00	11.00±0.00	11.00±0.00
whole sequence	108.88±2.54	109.19±1.58	121.10±4.63	125.43±4.10

Table B.6: Sequence length of observed and generated sequences in test set by regions and chain type.

B.2.3 Conditional Generation Recovers Pairing Sequences

t-SNE plots on sequence representation are similar to those without sequence clustering (Figure B.3a, B.3b and B.3c). When comparing on recovery rate of target sequences, we found that cluster split leads to slightly stronger bias towards specific families (Figure B.5). Sequence recovery is similar to that without sequence clustering (Figure A.9 and B.6).



(a) Heavy and light chains (b) Gene loci in human (c) IGHV gene families in human

Figure B.3: t-SNE plot of encoder hidden states of test set sequences in progressively fine categories (chain types, gene loci, and gene families).

Figure B.4: t-SNE plot of antibody embeddings colorized by ANARCI annotated species

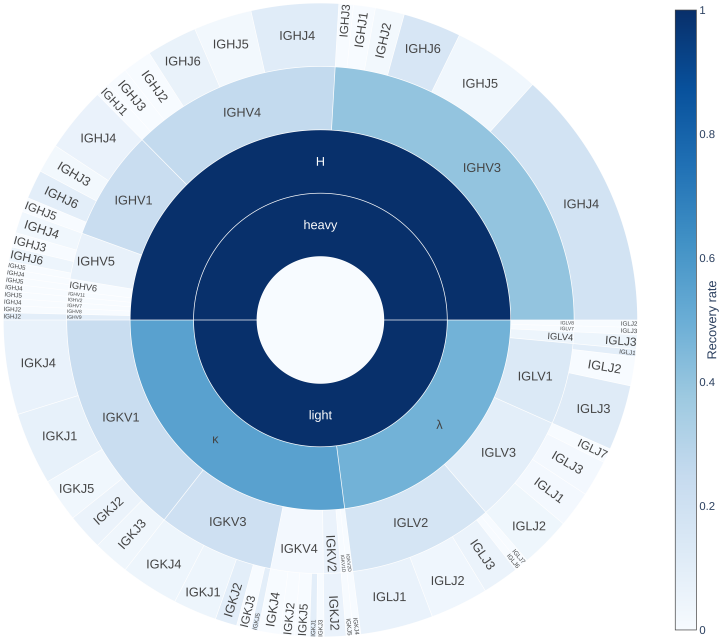


Figure B.5: Recovery rate of target chain type, gene loci, and gene families in sequence generation. Performance is represented in a hierarchical order, where parent classes are centered while children categories are on the periphery. On each rim, the arc lengths of categories are proportional to their populations in test set. Dark blue represents perfect recovery whereas white color implies low recovery rate.

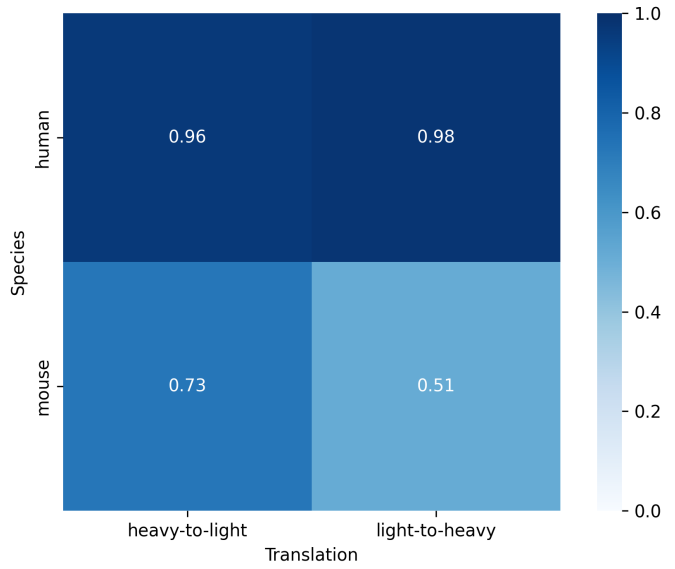


Figure B.6: Recovery rate on species by original species and translation direction.

B.2.4 Zero-shot Prediction from Paired Antibody Perplexity

Trained on clustered sequences, our model performs more weakly (p-value < 0.05) on one dataset. Results are largely unaffected by sequence clustering.

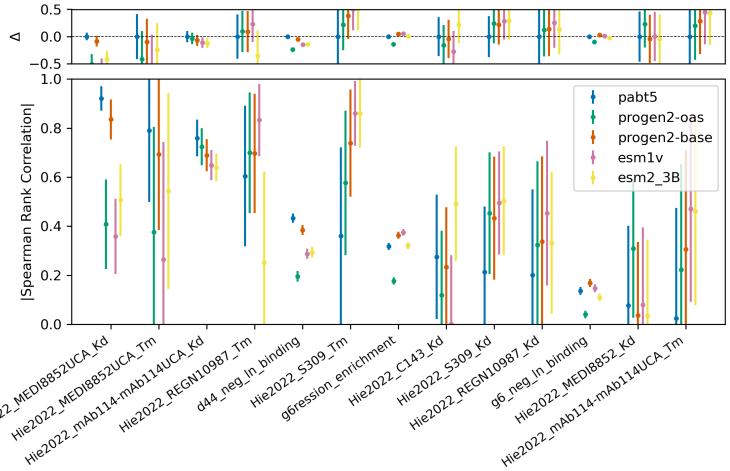


Figure B.7: Zero-shot prediction performance on antibody measurements of our model and state-of-the-art on all datasets. x-axis represents datasets. (Top) The difference in absolute spearman rank correlation (SRC) between our model and state-of-the-art. (Bottom) Absolute SRC between model (pseudo-)perplexity and measurements. Error bars are estimated in standard deviation with 1000 bootstrap samples.

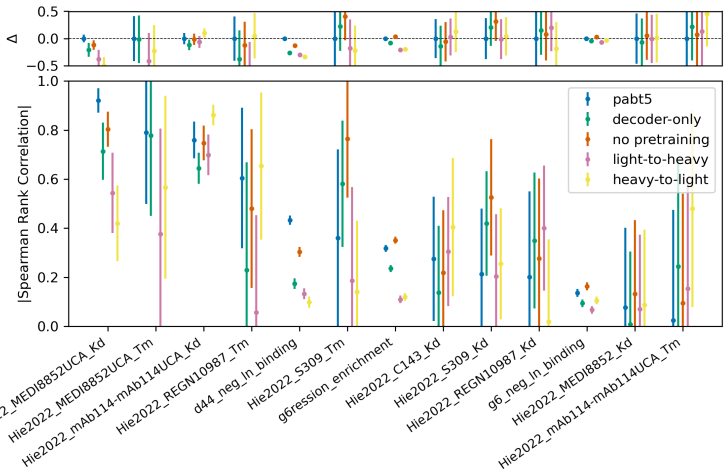


Figure B.8: Ablation study on zero-shot prediction on all datasets. x-axis represents datasets. (Top) The difference in absolute spearman rank correlation (SRC) between our model and ablation. (Bottom) Absolute SRC between model (pseudo-)perplexity and measurements. Error bars are estimated in standard deviation with 1000 bootstrap samples.