Stable Online and Offline Reinforcement Learning for Antibody CDRH3 Design

Yannick Vogt[1,2,*]**, Mehdi Naouar**[1,2]**, Maria Kalweit**[1,2]**, Christoph Cornelius Miething**[2,3]**,
Justus Duyster[3]**, Roland Mertelsmann**[2,3]**, Gabriel Kalweit**[1,2,†]**, Joschka Boedecker**[1,2,4,†]
[1]University of Freiburg, [2]Collaborative Research Institute Intelligent Oncology (CRIION),
[3]University Medical Center Freiburg, [4]BrainLinks-BrainTools

Abstract

The field of antibody-based therapeutics has grown significantly in recent years, with targeted antibodies emerging as a potentially effective approach to personalized therapies. Such therapies could be particularly beneficial for complex, highly individual diseases such as cancer. However, progress in this field is often constrained by the extensive search space of amino acid sequences that form the foundation of antibody design. In this study, we introduce a novel reinforcement learning method specifically tailored to address the unique challenges of this domain. We demonstrate that our method can learn the design of high-affinity antibodies against multiple targets *in silico*, utilizing either online interaction or offline datasets. To the best of our knowledge, our approach is the first of its kind and outperforms existing methods on all tested antigens in the Absolut! database.

1 Introduction

Antibody-based therapeutics have revolutionized the treatment of various conditions [1–3]. Especially for highly individual diseases like cancer, antibody-therapeutics offer great potential in guiding the cellular immune response. However, the process of designing *de novo* antibodies that bind effectively to given targets poses substantial challenges, primarily due to the combinatorial explosion of potential amino acid sequences that need to be considered. Research has shown that the CDRH3 region's diversity within an antibody suffices for most antibody specificity [4]. Consequently, designing CDRH3 sequences is typically used as a surrogate for the design of entire antibodies, as this reduces the search space significantly. Still, the search space for CDRH3 regions consisting of a combination of twenty naturally occurring amino acids (AAs) grows exponentially with 20^L, where L is the length of the CDRH3 regions.

In recent years, Reinforcement Learning (RL) methods [5] have demonstrated their efficacy in hard exploration tasks and search spaces with high branching factors, as exemplified by AlphaGO [6], and learning from pre-collected datasets [7]. In addition, transformer-based language models like GPT [8, 9] have exhibited remarkable advancements in the domain of natural language processing. Leveraging the analogy between natural language and protein amino acid sequences, these models have proven valuable for protein-related tasks [10]. Taking up on these two recent developments, our method combines RL with transformer-based neural networks to effectively design binding antibody CDRH3 sequences *in silico*.

Applying RL in this domain, however, appears challenging, as current sophisticated methods typically rely on huge datasets acquired through an agent's active engagement and direct interaction with its environment. Consequently, a method that can learn from pre-collected datasets would offer a

---

[*]Corresponding author: vogty@cs.uni-freiburg.de
[†]These authors contributed equally.

great advantage in allowing for consecutive reuse of previous trials. This is why we place particular emphasis on the *offline* setting. Offline RL in turn often faces challenges related to overestimation and stability [7]. Combining insights from Lan et al. [11], An et al. [12], and Ball et al. [13], we tackle overestimation by training an ensemble of Q-functions employing LayerNorm to approximate the binding energy of a given AA sequence. To address the issue of the high branching factor, we additionally employ an efficient implementation for *oversampling* of good candidates and shape the immediate reward to make small differences more apparent.

Our agent aims at minimizing the binding energy or maximizing the *affinity* between candidate antibody and target antigen, representing their strength of interaction, as this is closely tied to its therapeutic efficacy [14]. The binding energy in turn depends on many physical factors emerging from the structure and binding locations of candidate and target. To evaluate the affinity of antibody-antigen complexes *in silico* efficiently, we rely on the Absolut! software [2]. Complexes are thereby discretized onto a lattice representation using their alpha-carbon locations since current methods utilizing continuous structure representations have shown to be error-prone or too time-consuming [15, 16].

Two methods benchmarked on Absolut! are Structured Q-learning (SQL) [17] and AntBO [18]. The first shares similarities with our approach as it utilizes RL principles and employs an attention-based neural network. However, the authors observed that designing sequences step-by-step resulted in suboptimal binders, whereas generating the sequence as a whole improved performance. In contrast, our method adopts a step-by-step approach allowing the agent to take incomplete sequences into account, resulting in even better binders, and is directly applicable to the offline setting. The second is based on Bayesian Optimization using Gaussian processes, demonstrating high sample efficiency. However, the cubic complexity of Gaussian processes in relation to the number of samples limits the scalability of such approaches.

Our contributions are threefold. First, we introduce the (to the best of our knowledge) first offline RL method for antibody design, moving towards real-world applicability in this domain. Second, we show that our method is applicable to both the online and offline settings without any modification. Third, we introduce a new state-of-the-art on all tested antigens in the Absolut! benchmark.

2 Background

In this section, we briefly introduce the necessary background for our method and experiments.

Markov Decision Process In the following, we describe how to model the task of designing a CDRH3 region as a Markov Decision Process (MDP).

Generally, we consider an MDP for deterministic fixed horizon systems as a tuple $\langle S, A, T, R, s_0 \rangle$, where S is the set of states, A is the set of actions, and $T : S \times A \to S$ is the deterministic transition function, where $T(s, a)$ represents the state that the MDP will transition to from state s under action a. $R : S \times A \to \mathbb{R}$ is a reward function, where $R(s, a)$ represents the immediate reward r received after taking action a in state s. The initial state of the MDP is represented by $s_0 \in S$.

In our setting, the initial state s_0 is given as a sequence of blank tokens $[[B_i]_{i=0}^{L-1}]$, representing the absence of any AAs, where L is the desired length of the sequence. The Absolut! software, used to evaluate designed CDRH3 sequences, limits the length of a CDRH3 to $L = 11$ AAs. All other states are then derived from the initial state by replacing the tokens starting from B_0 to B_{10}. The transition function, therefore, transforms a state s_t to a next state s_{t+1} where B_t is replaced by the current action $a_t \in A$ where A is the set of twenty naturally occurring AAs, resulting in $[[a_i]_{i=0}^{t}, [B_j]_{j=t+1}^{L-1}]$. The reward function R returns zero for all states and actions except for the a_{10} which completes the CDRH3 sequence. There, the reward is derived from the binding energy computed by the Absolut! software for the designed CDRH3 and the given target antigen.

Online and Offline Reinforcement Learning The goal of an RL agent interacting with an MDP is to learn a policy $\pi : S \to A$, which maps each state to an action, that maximizes the expected sum of future rewards until a fixed terminal time step $L - 1$, defined as $\sum_{t=0}^{L-1} R(s_t, a_t)$, where s_t is the state at time t and a_t is the action taken at time t under policy π. We approximate this sum using a Q-function $Q : S \times A \to \mathbb{R}$ represented by a neural network. Consequently, the policy π is derived from Q as $\pi(s) = \arg\max_a Q(s, a)$.
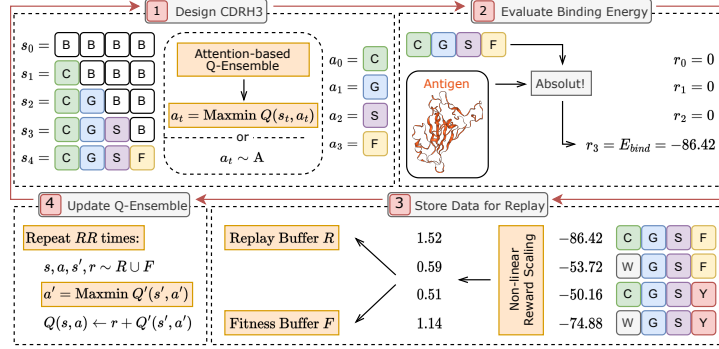
Figure 1: Visualization of our method on fictive CDRH3 sequences of length four. Our method repeatedly designs CDRH3 sequences (1), and evaluates them using the Absolut! software (2), stores the gathered data (3), and updates its Q-function and thereby policy (4). We highlight important components of our method in orange.

The field of RL divides into two popular sub-paradigms: *online* and *offline* RL. The more common *online* RL describes agents learning from self-executed actions in the environment, whereas *offline* methods describe agents learning solely from pre-existing datasets. The latter gained more and more traction in recent years, especially in domains where data collection can be tedious or dangerous [7]. We, therefore, consider the offline setting more feasible for antibody design, where the exploitation of growing and older datasets instead of time-intensive online evaluations appears significant and could prove highly valuable. In both paradigms and our domain model-based RL can be considered [19, 20]. We refrain from model usage to omit possible exploitation or bias of the learned model [21].

3   Stable Reinforcement Learning for Antibody Design

In the following, we explain the components of our off-policy RL algorithm. Further, we visualize our method and its interaction with the Absolut! software in Fig. 1. We provide pseudocode, the full network architecture, and additional details in the Appendix.

Favor Exploration – Replay the Fittest   In proteins, the effect of a mutation depends on the protein sequence as a whole. Therefore, two changes in the sequence which, isolated, each reduce the protein's utility can have a synergistic positive effect when they occur at the same time. This effect, which is called epistasis, can prevent proteins from improving [22]. We hypothesize that a similar effect may be prevalent when an antibody-designing RL agent emphasizes exploitation over exploration, resulting in only single changes. We thus advocate for a high number of random actions sampled from the action space, $a \in A$, during training.

To counterbalance reduced exploitation and to over-sample the very best sequences during replay, we introduce a *Fitness Buffer*. Implemented as a min-heap, this buffer contains transitions corresponding to the k best sequences discovered. Intuitively, this additional buffer facilitates learning from infrequently occurring sequences. During replay, the agent is trained on a mixture of samples from the Fitness Buffer and a conventional Replay Buffer which stores all observed transitions (cf. Algorithm 1 in the Appendix). This technique can be seen as an adaption of rank-based Prioritized Experience Replay [23] where the importance of a transition corresponding to one of the best sequences is multiple times higher than that of a common sequence.

Stabilizing Reinforcement Learning with Ensembles   When dealing with large search spaces, such as protein sequences, it is infeasible to explore each possibility. Therefore, there will always be actions that are out of the training distribution of the RL agent for a given state. The value of such actions is often erroneously overestimated [24, 7]. We approach this issue by utilizing an ensemble of N learned Q-functions as in Maxmin Q-learning [11] and many offline RL algorithms [12], and define the Maxmin-policy as $\mathrm{Max\,min}\, Q(s, a) := \arg\max_{a \in A} \min_{i \in \{1, \dots, N\}} Q_i(s, a)$. We hypothesize that this design choice allows us to apply the agent to both online and offline RL.

3

Table 1: Final binding energy per method and antigen. Shown is our method in the online and offline setting, as well as SQL and AntBO. We mark results significantly outperforming AntBO with [1] and results significantly outperforming both AntBO and SQL with [2]. Absolut! and Online Lead represent the best energy contained in the Absolut! database and offline dataset (collected using an online agent) respectively. Mean and standard deviation are given over eight and ten seeds in our method and SQL respectively.

Antigen	Ours Online	SQL	AntBO	Absolut!	Ours Offline	Online Lead
3RAJ_A	-123.09 ± 1.74^{1}	-123.00 ± 2.88	-119.07	-116.74	-125.08 ± 0.09^{2}	-125.19
2DD8_S	-131.06 ± 1.36^{1}	-129.00 ± 4.10	-127.57	-125.07	-132.37 ± 0.11^{1}	-132.45
1ADQ_A	-112.55 ± 0.06^{2}	-112.00 ± 0.41	-110.70	-108.53	-112.56 ± 0.06^{2}	-112.58
1OB1_C	-111.27 ± 0.98^{1}	-111.00 ± 0.51	-108.78	-108.78	-111.78 ± 0.08^{2}	-111.88
2YPV_A	-116.48 ± 0.91^{1}	-115.00 ± 2.36	-108.75	-114.47	-117.30 ± 0.08^{2}	-117.37
1NSN_S	-107.34 ± 0.35^{2}	-106.00 ± 1.70	-105.02	-107.16	-107.65 ± 0.06^{2}	-107.75
1WEJ_F	-88.66 ± 0.58^{1}	-88.30 ± 0.13	-86.89	-86.03	-89.99 ± 0.23^{2}	-90.11
2JEL_P	-88.15 ± 1.44^{2}	-84.70 ± 1.62	-84.33	-86.05	-90.54 ± 0.03^{2}	-90.57

Attention-based Q-Networks   Similar to SQL and large language models [8], which can process natural language, we use a Q-network based on multi-head self-attention to process a sequence of amino acids. To facilitate the training of such an architecture we propose using a high Replay Ratio (RR) [25], meaning many updates to the network per environment interaction.

Reward Scaling for Binding Energy   To amplify the difference in binding energy values, we non-linearly scale the reward to $r_{scaled} = (\frac{r}{\beta})^2$. Thereby, β is a constant to reduce the magnitude of the rewards. This amplifies both the difference between weak binders and strong binders as well as the difference between strong binders where binding energy values may differ only minimally, often down to mere fractions of a unit. We, therefore, argue that this modulation of the immediate reward facilitates the noisy learning progress of neural networks in RL.

4   Results and Discussion

In this section, we present and discuss the results of our experiments both in the online and offline settings. To enable a comparison, we evaluate our method on the same antigens as SQL and AntBO. The hyperparameter settings, as well as an analysis of statistical significance and an ablation study, are given in the Appendix.
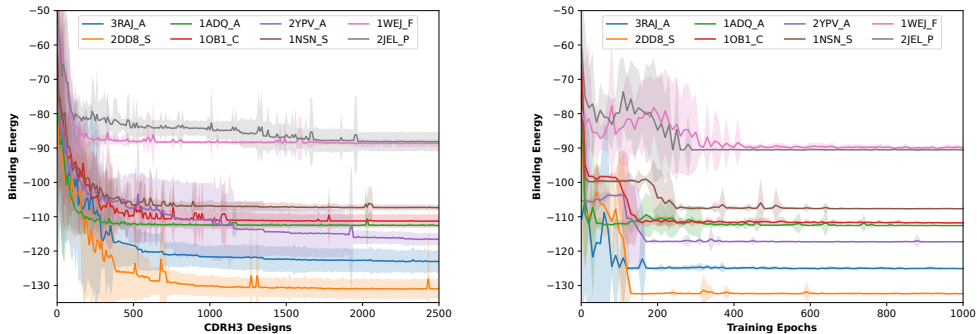

Figure 2: Progress of the binding energy of our method on different antigens, colored as displayed in the provided legend. The left plot depicts the online setting, while the offline setting is shown in the right plot. Mean and two standard deviations over eight seeds are shown.

Online Learning   In Table 1 we report the binding energy reached with our method, as well as SQL [17], and AntBO [18]. Further, we report the lowest binding energy of 6.9 million murine

CDRH3 sequences in the Absolut! database. We observe that our method outperforms or matches binding energies found by existing methods on all tested antigens and exceeds the best values in the Absolut! database. Note, that we use more CDRH3 design steps during our experiments than SQL and AntBO. In Fig. 2 we visualize the evaluation progress of our method on different antigens. Notably, the evaluation progress is stable and does not collapse.

CDRH3 Analysis   Our method designs CDRH3 sequences with the sole objective of maximizing the affinity to a given antigen. Nonetheless, we analyze the AA distribution and binding specificity of designed sequences binding 1ADQ_A to derive further insights. Thereby, we define a sequence as binding if it reaches the affinity of the top 1% of 6.9 million murine CDRH3 sequences on the corresponding antigen in the Absolut! database. We define a CDRH3 sequence as specific if it binds only one of the eight antigens [26]. The logo plots shown in Fig. 3 illustrate the AA distributions of high-affinity CDRH3 sequences, revealing a bias toward leucine, phenylalanine, and more generally AAs with hydrophobic side chains. This coincides with the findings by Cowen-Rivers et al. [17]. Additionally, we noted that the fraction of specific binders in the top 0.01% CDRH3 sequences in the Absolut! database is significantly higher than the fraction of specific binders, of equal or higher affinity, discovered in our experiments. We omit a more detailed analysis of the designed CDRH3 sequences regarding their biophysical properties as those are neither contained in Absolut! software nor optimized by our method. We suggest future work regarding antigen-specific CDRH3 design and the development of simulators incorporating more biophysical properties.
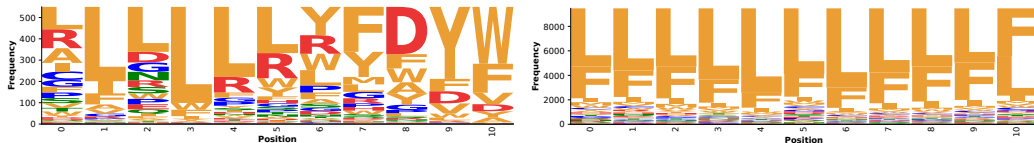

Figure 3: Amino acid frequency per position in the top 0.01% CDRH3 sequences in the Absolut! database (left) and in CDRH3 sequences, reaching at least the same affinity, discovered in our experiments (right). Colors indicate the AA polarity.

Offline Learning   To evaluate our method's performance in the offline setting we copy the Replay and Fitness Buffers of the offline RL agents from a single online agent and train them without environment interaction. In Table 1 we present the binding energies reached by our offline agents as well as the best energy contained in their Replay Buffer (named Online Lead). We observe that our agents can closely match this peak affinity and exhibit low standard deviations despite never interacting with the MDP. In Fig. 2 we display the evaluation progress of our offline agents. It is evident that the majority of offline agents converge significantly faster and demonstrate similar or enhanced stability compared to their online counterparts. It is noteworthy that our method was not altered in any way to support offline learning. In Section F in the Appendix, we show preliminary results on how to use a trained offline agent to generate new CDRH3 sequences with improved energy distribution compared to the offline dataset.

5   Conclusion

We introduced a novel RL method that addresses the unique challenges of the antibody design task. Through experiments conducted on the Absolut! software, we demonstrated its stability and effectiveness in designing high-affinity antibody CDRH3 sequences, surpassing the performance of existing methods. Our method has the advantage of being applicable to both online and offline RL settings, enabling learning from pre-collected datasets. This capability has significant potential for real-world experiments since performing real-time evaluation of antibodies for online training is infeasible. Further work is recommended to expand our method to include the design of antigen-specific CDRH3 sequences.

Acknowledgments

References

[1] Richard A Norman, Francesco Ambrosetti, Alexandre M J J Bonvin, Lucy J Colwell, Sebastian Kelm, Sandeep Kumar, and Konrad Krawczyk. Computational approaches to therapeutic antibody design: established methods and emerging trends. *Briefings in Bioinformatics*, 21(5), 10 2019. doi: 10.1093/bib/bbz095.

[2] Philippe A. Robert, Rahmad Akbar, Robert Frank, Milena Pavlović, Michael Widrich, Igor Snapkov, Andrei Slabodkin, Maria Chernigovskaya, Lonneke Scheffer, Eva Smorodina, Puneet Rawat, Brij Bhushan Mehta, Mai Ha Vu, Ingvild Frøberg Mathisen, Aurél Prósz, Krzysztof Abram, Alex Olar, Enkelejda Miho, Dag Trygve Tryslew Haug, Fridtjof Lund-Johansen, Sepp Hochreiter, Ingrid Hobæk Haff, Günter Klambauer, Geir Kjetil Sandve, and Victor Greiff. Unconstrained generation of synthetic antibody-antigen structures to guide machine learning methodology for real-world antibody specificity prediction. *bioRxiv*, 2022. doi: 10.1101/2021. 07.06.451258.

[3] Hélène Kaplon, Silvia Crescioli, Alicia Chenoweth, Jyothsna Visweswaraiah, and Janice M. Reichert. Antibodies to watch in 2023. *mAbs*, 15(1), 2023. doi: 10.1080/19420862.2022. 2153410.

[4] John L Xu and Mark M Davis. Diversity in the cdr3 region of vh is sufficient for most antibody specificities. *Immunity*, 13(1), 2000. doi: https://doi.org/10.1016/S1074-7613(00)00006-6.

[5] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6.

[6] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676), Oct 2017. doi: 10.1038/nature24270.

[7] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[8] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[10] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *CoRR*, abs/2301.06568, 2023. doi: 10.48550/ARXIV.2301.06568.

[11] Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[12] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

[13] Philip J. Ball, Laura M. Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023.

[14] Emily K. Makowski, Patrick C. Kinnunen, Jie Huang, Lina Wu, Matthew D. Smith, Tiexin Wang, Alec A. Desai, Craig N. Streu, Yulei Zhang, Jennifer M. Zupancic, John S. Schardt, Jennifer J. Linderman, and Peter M. Tessier. Co-optimization of therapeutic antibody affinity and specificity using machine learning models that generalize to novel mutational space. *Nature Communications*, 13(1), Jul 2022. doi: 10.1038/s41467-022-31457-3.

[15] Gwen R Buel and Kylie J Walters. Can alphafold2 predict the impact of missense mutations on structure? *Nature structural & molecular biology*, 29(1), 2022.

[16] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637), 2023. doi: 10.1126/science.ade2574.

[17] Alexander I. Cowen-Rivers, Philip John Gorinski, Aivar Sootla, Asif Khan, Furui Liu, Jun Wang, Jan Peters, and Haitham Bou-Ammar. Structured q-learning for antibody design. *CoRR*, abs/2209.04698, 2022. doi: 10.48550/ARXIV.2209.04698.

[18] Mohammad Asif Khan, Alexander I. Cowen-Rivers, Derrick-Goh-Xin Deik, Antoine Grosnit, Kamil Dreczkowski, Philippe A. Robert, Victor Greiff, Rasul Tutunov, Dany Bou-Ammar, Jun Wang, and Haitham Bou-Ammar. Antbo: Towards real-world automated antibody design with combinatorial bayesian optimisation. *CoRR*, abs/2201.12570, 2022.

[19] Christof Angermüller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy J. Colwell. Model-based reinforcement learning for biological sequence design. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[20] Moksh Jain, Emmanuel Bengio, Alex Hernández-García, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with gflownets. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

[21] Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 2021.

[22] Nicholas C Wu, Lei Dai, C Anders Olson, James O Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5, jul 2016. doi: 10.7554/eLife.16965.

[23] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[24] Hado van Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*. Curran Associates, Inc., 2010.

[25] Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G. Bellemare, and Aaron C. Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[26] Chun Zhou and Serge Przedborski. Intrabody and parkinson's disease. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1792(7), 2009. doi: https://doi.org/10.1016/j.bbadis.2008.09.001.

[27] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540), 2015. doi: 10.1038/NATURE14236.

A Architecture

In the following, we give the details of our neural network architecture, which we visualize in Fig. 4.
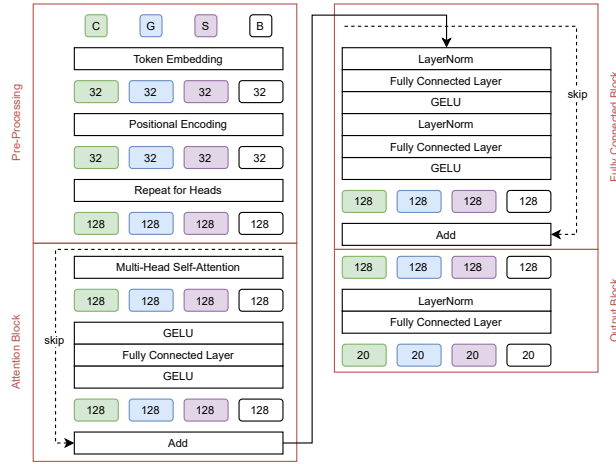

Figure 4: Visual summary of our network architecture.

On a high level, the architecture consists of pre-processing of the sequence, followed by an attention block, a fully connected block, and an output block. We add skip connections around the attention block and the fully connected block.

Pre-processing: We begin by embedding the discrete amino acid tokens into a learnable continuous 32-dimensional vector. We add a 32-dimensional learnable positional embedding to allow differentiating the same amino acid type at different positions in the sequence. We then repeat the embeddings four times, once per attention head.

Attention block: We apply LayerNorm, followed by multi-head self-attention using four heads. We add a look-ahead preventing mask to the attention logits before applying a softmax. This mask is implemented as a triangular matrix filled with $-\infty$. Therefore, the attention for tokens following the current token will become zero. After applying the attention to the value-heads output, we apply a GELU activation. As we utilize four heads, the resulting latent dimension is a 128-dimensional vector for each token in the sequence. We project this latent representation using a fully connected layer, without LayerNorm as this would normalize the four different attention outputs, followed by another GELU activation. As we do not repeat this self-attention block, our architecture is not strictly a transformer but rather an attention-based fully connected network.

Fully connected block: We repeat two iterations of LayerNorm, fully connected layers, and GELU activation, without altering the latent dimension.

8

Output block: We apply a final LayerNorm before mapping the latent representation to a 20-dimensional output, representing Q-values for the twenty naturally occurring AAs. To derive the AA to be placed at a specific token position, we evaluate the Q-values of the respective B token.

B Method Summary

In Algorithm 1 we present an overview of our method. The fraction of samples from the Fitness Buffer is determined by the fitness fraction f. Note, that this pseudocode depicts the online setting. For the offline setting lines 4-12 can be removed and B and F need to be initialized from a dataset.

Algorithm 1 Stable Reinforcement Learning for Antibody Design

---

1: **Input:** ensemble size N, exploration probability ϵ, reward scaling constant β, Fitness Buffer size k, replay ratio RR, batch size b, fitness fraction f, learning rate α, target update ratio τ,
2: **Initialize:** N Q-networks $Q_{0,...,N-1}$ and target networks $Q'_{0,...,N-1}$, Replay Buffer B, Fitness Buffer F of size k
3: **for** $e = 0$ **to** 2499 **do**
4:    s_0= initial state of the MDP
5:    **for** $i = 0$ **to** 10 **do**
6:        $Q_{min}(s,a) \leftarrow \min_{i \in \{1,...,N\}} Q_i(s,a), \forall a \in A$
7:        get ϵ-greedy action a_i from Q_{min}
8:        Execute action a_i, receive r_i, s_{i+1}
9:        apply non-linear scaling $r_i \leftarrow (\frac{r_i}{\beta})^2$
10:    **end for**
11:    Store trajectory $(s_i, a_i, r_i, s_{i+1}) \forall i \in [0, 10]$ in B
12:    pushpop trajectory $(s_i, a_i, r_i, s_{i+1}) \forall i \in [0, 10]$ onto F
13:    **for** 0 **to** $RR - 1$ **do**
14:        sample minibatch of size b with transitions (s_D, a_D, r_D, s'_D) from $B \cup F$ with $f\%$ from F
15:        $y = \max_{a' \in A} Q'_{min}(s'_D, a')$
16:        Update all Q-functions $Q_i(s_D, a_D) \leftarrow r_D + y$ using gradient descent with learning rate α
17:    **end for**
18:    Soft-update target network weights using τ
19: **end for**

C Hyperparameters

Due to limited computational resources, we did not perform extensive hyperparameter sweeps but instead adapted the hyperparameters to optimize results in preliminary runs. We assume that our method could benefit from a better choice of hyperparameters.

The ensemble size N was set to 10 in the online setting while being set to 20 in the offline setting to achieve higher stability. The remaining hyperparameters are shared for all experiments and are set as follows. Exploration probability (ϵ) = 0.2, Reward scaling constant (β) = 70, Fitness Buffer size (k) = 10, Replay Ratio (RR) = 50, Batch size (b) = 64, Fitness fraction (f) = 20%, Learning Rate (α) = 0.0005, and Target Update Ratio (τ) = 0.002.

D Statistical Significance

To evaluate the significance of our improvements over other methods we utilize Welch's t-test on the final binding energy. As in Section 4, the values for SQL were taken from the respective publication. However, the authors only provided integer-valued, presumably rounded results for some targets. The values for AntBO are derived from the published normalized values by scaling them with the normalization values given in the Absolut! column of Table 1.

P-values resulting from Welch's t-test give the probability that the null hypothesis, that is, the final energy scores of the two evaluated methods are drawn from equal distributions, holds true. We reject the null hypothesis for probabilities below 5%. When evaluating against SQL we utilize the mean

and standard deviations given over ten seeds and the mean and standard deviation of our method over eight seeds. Note, that the presumably rounded values in the SQL publication distort our analysis. As no standard deviation for AntBO was published, we employ a one-sample t-test utilizing the provided mean value. Given the available data, these tests allow, to the best of our knowledge, the least biased evaluation.

We find that our method in the online setting gives a significant improvement compared to AntBO on all evaluated antigens. Despite reaching higher binding affinity than SQL on all eight targets, the improvement is only significant on $\frac{3}{8}$ of the targets, namely 1ADQ_A, 1NSN_S, and 2JEL_P. In the offline setting our method again gives a significant improvement compared to AntBO on all eight antigens. Compared to SQL we observe a significant improvement on all antigens except 2DD8_S.

E   Ablation Study

To validate the effect of our chosen components we carried out multiple ablation studies on the antigen 2DD8_S and further compared our method to Deep Q-networks (DQN) Mnih et al. [27] with the same network architecture and hyperparameter setting. We refer to our method with all its components as "Ours". We evaluated removing all components (w/o all), the Fitness Buffer (w/o Fitness), reducing the RR to 10 (w/o RR), reducing the RR while removing the Fitness Buffer (w/o RR&Fitness), and removing the nonlinear reward scaling (w/o Scaling). On the left side of Fig. 5, we visualize the convergence of our method in comparison to DQN both with and without all our components. On the right side, we visualize the effect of removing a subsection of the components. We can observe that all components have an influence on the training progress, especially in the early training stage. Further, we see that DQN did not converge within 2500 training iterations, neither with nor without our introduced components. This shows the effect of the Maxmin-Ensemble, as our method could be described as DQN utilizing said ensemble instead of a plain neural network.
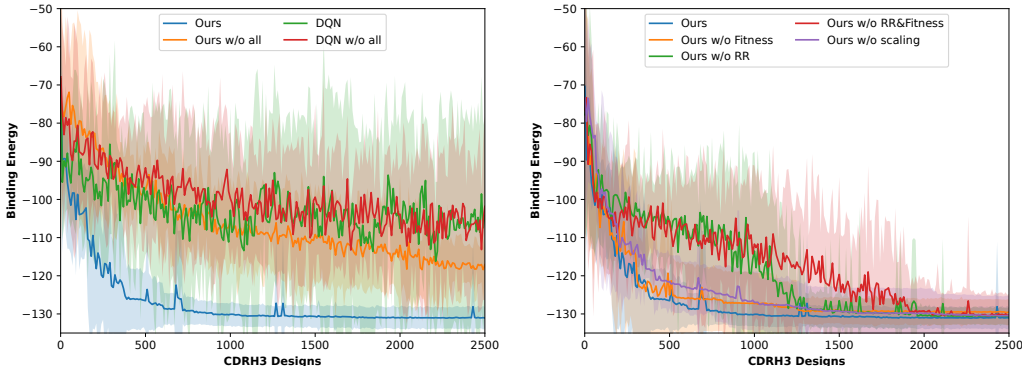

Figure 5: Progress of energy scores of our method and different ablation runs on the antigen 2DD8_S. Mean and two standard deviations over eight seeds are displayed. "Ours" represents our method with all its components. We refer to our method without a high RR, Fitness Buffer, and Scaling as "w/o all" as that reflects our method without all its components. On the left, we visualize the convergence of our method in comparison to DQN both with and without all our components. On the right, we visualize the effect of removing a subsection of the components.

F   Stochastic CDRH3 Design

In this section, we present preliminary results. We utilize a trained ensemble of Q-networks to generate novel CDRH3 sequences. In contrast to the deterministic Maxmin-policy we utilized in our online and offline experiments, we here define a stochastic policy based on a mixture of the deterministic $\arg\max$ and a gumble-softmax distribution based on normalized Q-values which we refer to as the softmax policy. This approach can theoretically be implemented with Q-networks from both online and offline settings and could also be used for exploration. Here, we test it with

an ensemble of Q-networks trained offline on CDRH3 sequences explored on the antigen 2DD8_S. To verify that using the softmax policy has a positive effect, we also added a control distribution which we generated by mixing the deterministic $\arg\max$ and a uniform distribution over AAs called uniform-policy. In Fig. 6 we visualize the density distributions over binding energies, showing an increase in high-affinity binders using the softmax sampling. The distribution's mean and standard deviation shift from -107.04 ± 9.85 to -108.55 ± 9.55 and -11.76 ± 10.75 using the uniform and softmax policy respectively. All those shifts are significant with p-values below $1e-5$. Using the softmax policy we further generated multiple new best binders outside the training distribution. Note, that this distribution is sensible to the fraction of AAs derived using the $\arg\max$. A higher fraction leads to higher average affinity but also more duplicated CDRH3 sequences in the set of newly generated sequences. Further, we found that out of ~ 1500 unique CDRH3 sequences generated using this approach, only ~ 100 were contained in the training distribution, showing that this approach can be used to generate new CDRH3 distributions with improved energy distributions.
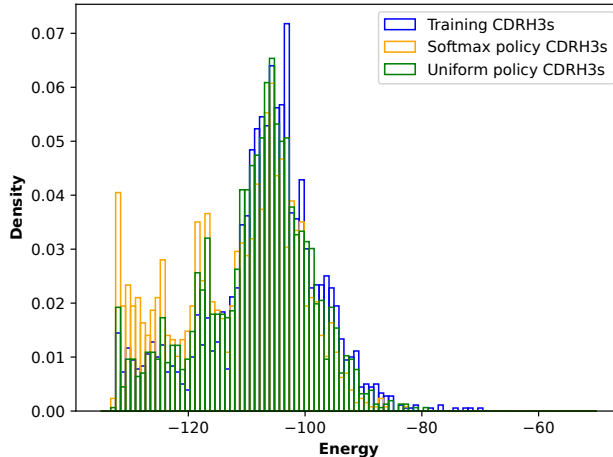

Figure 6: Energy distributions over the offline training dataset as well as newly generated CDRH3 sequences using the softmax policy and uniform policy. Duplicated CDRH3 sequences have been removed.