Is Sequence Information All You Need for Bayesian Optimization of Antibodies?

Sebastian W. Ober
BigHat Biosciences
sober@bighatbio.com

Calvin McCarter
BigHat Biosciences

Aniruddh Raghu
BigHat Biosciences

Yucen Lily Li
NYU

Alan N. Amin
NYU

Andrew Gordon Wilson
NYU

Hunter Elliott
BigHat Biosciences
helliott@bighatbio.com

Abstract

Bayesian optimization is a natural candidate for the engineering of antibody therapeutic properties, which is often iterative and expensive. However, finding the optimal choice of surrogate model for optimization over the highly structured antibody space is difficult, and may differ depending on the property being optimized. Moreover, to the best of our knowledge, no prior works have attempted to incorporate structural information into antibody Bayesian optimization. In this work, we explore different approaches to incorporating structural information into Bayesian optimization, and compare them to a variety of sequence-only approaches on two different antibody properties, binding affinity and stability. In addition, we propose the use of a protein language model-based "soft constraint," which helps guide the optimization to promising regions of the space. In doing so, we explore a wide range of ways in which generative modeling can be incorporated into Bayesian optimization for antibodies. We find that certain types of structural information improve data efficiency in early optimization rounds for stability, but have equivalent peak performance. Moreover, when incorporating the protein language model soft constraint we find that the data efficiency gap is diminished for affinity and eliminated for stability, resulting in sequence-only methods that match the performance of structure-based methods, raising questions about the necessity of structure in Bayesian optimization for antibodies.

1 Introduction

Therapeutic antibodies are an important class of drugs that are rapidly increasing in popularity for the treatment of a wide range of challenging diseases [1]. To develop a successful antibody for therapeutic purposes, it is not only necessary that it binds to its target at the desired strength, but it must also have satisfactory "developability" properties [2]: for instance, the antibody must be thermostable, have low hydrophobicity, and express well. Structure-based diffusion generative models have proven useful in non-iterative antibody engineering [3] and *de novo* antibody design [4], yet in general yield antibodies still requiring further refinement to meet the criteria of a valid therapeutic. In order to satisfy these criteria, it is common to require numerous rounds of iterative optimization using wet lab experiments. The relatively small datasets involved and the high cost of wet lab experiments form an ideal setting for a technique such as Bayesian optimization [5], which attempts to optimize these properties in an uncertainty-guided way. However, successfully applying Bayesian optimization requires a number of difficult design choices, most notably in the choice of surrogate model and acquisition function.

A number of works have attempted to elucidate some of these choices, either by proposing new methods [e.g., 6, 7, 8], or by benchmarking different methods on relevant problems [9]. However, to the best of our knowledge, none of these works have attempted to incorporate *structural* information, which has been shown to be beneficial for non-Bayesian iterative antibody optimization [10]. Moreover, it is unclear which antibody properties structural information might be relevant for: developability properties intrinsic to the antibody, or binding properties, a function of the antibody-antigen (Ab-Ag) interaction. The latter is especially uncertain in the common scenario considered here, where the binding pose of the antibody to its target is not known, and cannot be reliably predicted [11].

In this work, therefore, we attempt to understand how best to incorporate structural information, in which situations it is helpful, and how it compares to sequence-only approaches in the low-data regime that is most amenable to Bayesian optimization. In particular, we aim to address the following questions:

- How can we incorporate structural information into Bayesian optimization surrogate models?
- For what tasks does structural information aid optimization?
- Does incorporating antibody-specific structural models boost performance over general protein models?
- Finally, is structural information *necessary* for good optimization performance?

Additionally, we propose a novel (to the best of our knowledge) means of incorporating sequence-only prior information through a protein language model "soft constraint," which we use to help elucidate answers to these questions. In answering these questions, we thereby explore the incorporation of a range of deep learning and generative models: structure prediction models [e.g., IgFold 12], inverse folding models [e.g., ProteinMPNN 13], protein foundation models [e.g., ESM-2 14], and antibody-specific protein language models [e.g., Sapiens 15].

2 Background

Bayesian optimization [BO 5] is a powerful uncertainty-aware framework for the optimization of expensive black-box functions. Given potentially noisy observations $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ of the target function $g : \mathcal{X} \to \mathbb{R}$, BO constructs an uncertainty-aware surrogate model f from the data. The surrogate model is then used in tandem with an acquisition function, $a : \mathcal{X} \to \mathbb{R}$, which is used to determine which point to query next. The choice of query point is typically achieved by balancing exploration and exploitation using the surrogate model's uncertainty. The point and its acquired value are then appended to $\mathcal{D}$, and the process is repeated until we exhaust our evaluation budget of the expensive function.

The choice of surrogate model and acquisition function is crucial to the success of BO. The most common model class for BO is the Gaussian process [GP 16]. Using a GP, we model the data as $y_i = f(\mathbf{x}_i) + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}\left(0, \sigma^2\right)$, where we have placed a GP prior on f, given by $f \sim \mathcal{GP}\left(\mu, k\right)$. Note that a GP is defined entirely by its mean function, $\mu : \mathcal{X} \to \mathbb{R}$, and its kernel function, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

Perhaps the most popular acquisition function for BO is expected improvement [EI 17], due to its simple closed-form expression for GPs and its strong empirical performance. However, EI by itself is inadequate for handling constraints on the search space, which is often a requirement for real-world use: for instance, for antibody development we need to optimize its properties subject to the constraint that it expresses well, which has to be learned. In order to address this, [18] derived the expected contrained improvement acquisition function,

$$a\left(\mathbf{x}\right) = \mathrm{PF}\left(\mathbf{x}\right) \mathrm{EI}\left(\mathbf{x}\right), \tag{1}$$

where $\mathrm{PF}\left(\cdot\right)$ is the learned probability of feasibility.

3 Methods

We now turn to describing the methods that we use for our comparison of sequence-only and structure-based Bayesian optimization of antibodies. We provide further mathematical details in Appendix B, and provide a comprehensive list of evaluated methods in App. Table 1.

3.1 Pareto-aware batch Bayesian optimization

In typical wet lab setups, we often wish to acquire data in batches, ranging in size from tens to hundreds of molecules. To enable these large batches, we use the recent qHSRI approach of [19]. This approach finds the Pareto front of predicted mean and standard deviations, and uses a Sharpe ratio "portfolio approach" to decide which variants to select for the batch, based off the predicted mean-standard deviation hypervolume.

As opposed to BO over continuous spaces, where powerful multi-start gradient-based optimizers are typically used, discrete spaces pose an additional challenge when it comes to acquisition function optimization. We therefore follow [20] and use a genetic algorithm. However, as we wish to find the predicted mean-standard deviation Pareto front, we turn to NSGA-II [21], a genetic algorithm designed specifically for Pareto-aware optimization. We refer the reader to App. B.1 for more details on the qHSRI acquisition function.

3.2 Sequence-only methods

Inspired by recent work [9, 22], we wish to include a strong sequence-only GP baseline. Following these works, we implement a Tanimoto kernel GP [23] with a constant mean function, using one-hot encodings as our baseline method, which we denote **OneHot-T**. Additionally, we use an encoding derived from the BLOSUM-62 matrix [24], which was shown to be effective in [9] (**BLO-T**). Finally, we use the mean-pooled embeddings derived from the ESM-2 650M model [14] as inputs to a Matérn-5/2 kernel GP (**ESM-M**).

As an additional baseline, we also compare against LaMBO [6], a complementary sequence-only approach that acquires sequences by using the latent space of a learned denoising autoencoder. Note that LaMBO has its own acquisition function, and is therefore the only method that we consider that does not use qHSRI acquisition.

3.3 Incorporating structural information

One of the main goals of our study is understanding in which scenarios structural information is useful in antibody property optimization. To that end, we investigate a number of options for incorporating such information. The first, and simplest, option is to use predicted structures (we assume we do not have access to ground truth structures, especially for each newly designed antibody we wish to consider) as direct inputs to a Matérn-5/2 kernel GP. To facilitate this, we predict the structure using IgFold [12], align the structure to the predicted parental structure, and extract (and flatten) the alpha-carbon coordinates as the encoding. We denote this model by **IgFold-M**. Note that this incorporates *only* 3D structural information without the identities of the specific amino acids comprising the protein and without reference to other known proteins or antibodies. We additionally consider two approaches that combine sequence and structure information. First, we concatenate these features to the ESM-2 features we use above as inputs, which we denote by **IgFold-ESM-M**. Second, we combine the IgFold-M kernel with the BLO-T kernel as a weighted sum kernel, which we denote by **IgFold-BLO-T**.

Finally, we consider the recently-proposed Kermut GP model [25], which attempts to combine structural information with sequence information. This model uses a parental structure (which we predict using IgFold), and ProteinMPNN [13] predictions from that structure alongside a sequence-only kernel to build a sequence-structure kernel for GP modeling. Note that this form of structural information implicitly compares to other known proteins, as ProteinMPNN was trained on a large set of known structures to estimate probabilities of specific amino acids given a structural motif. Additionally, Kermut uses zero-shot protein language model (pLM) predictions (defaulting to ESM-2 predictions) as part of its prior mean function. Note that we make some modifications to Kermut, which we ablate in App. C. We denote the final improved model by **Kermut-T**.

3.4 Incorporating antibody-specific information

Due to the particularities of antibodies (for instance, their combination of highly conserved frameworks with hyper-variable complementarity determining regions), we consider incorporating antibody-specific information into the Kermut model. In particular, we consider replacing the ProteinMPNN

predictions with those from AbMPNN [26], with the resulting method denoted as **AbMPNN-Kermut-T**. We explore various additional modifications in App. C.1.

3.5 Incorporating sequence prior information as a soft constraint

One major issue with many of the above pure-GP methods is that they do not incorporate prior information about the likelihood of an antibody sequence. Given the exponential size of antibody space, it would be wasteful to explore unlikely mutations. Moreover, without any information on the likelihood of certain mutations, BO will likely explore highly "unnatural" mutations that would cause the protein to fail to express or fold, leading to an inability to obtain meaningful property data.[1] To address this, and inspired by work on constrained BO from Eq. (1), we propose incorporating pLM predicted probabilities as a "soft constraint" on the acquisition function:

$$a_{\mathrm{pLM}}(\mathbf{x}) = \mathrm{p}_{\mathrm{pLM}}(\mathbf{x})\, a(\mathbf{x}), \tag{2}$$

where $\mathrm{p}_{\mathrm{pLM}}(\cdot)$ is the pLM's (pseudo)-likelihood of a sequence. In practice, we use the pseudo-likelihood from the Sapiens pLM [15] as a lightweight antibody-specific pLM for this purpose.

4 Experiments

We perform our experimental evaluation *in silico*, focusing on optimizing binding strength and stability using in-house oracles that are trained using data from a real-world optimization campaign, which we describe further in App. B.3. More specifically, we focus on optimizing an antibody's predicted dissociation constant (K_D) and melting temperature (T_m), starting with 50 examples taken from the early stages of the campaign. We perform nine acquisitions, aiming to acquire 80 molecules each iteration. However, in order to ensure the robustness of the BO algorithm, and to increase the fidelity of our *in silico* evaluation to the real world, we randomly drop 30 molecules each iteration: this could be representative of expression failures or other measurement failures that our oracles might not capture. Finally, we run each experiment three times, and plot the mean performance with standard error bars. Our experiment therefore hopefully captures meaningful differences between affinity and developability optimization, while remaining faithful to the low-data regime we wish to understand better.

4.1 Sequence-only methods

We first investigate the performance of sequence-only methods, without the use of any soft constraints. We compare the OneHot-T, BLO-T, and ESM-M models, along with LaMBO, in Fig. 1. We see that for affinity, the Tanimoto kernel models outperform the other methods: note that in the case of ESM-M, this is consistent with the result found in [9]. For T_m, we see that while ESM-M has strong initial performance, the Tanimoto kernel models catch up and result in largely equivalent final values. We note that LaMBO seemingly struggles in both settings, possibly due to the small initial dataset combined with the need to train a denoising autoencoder from scratch.

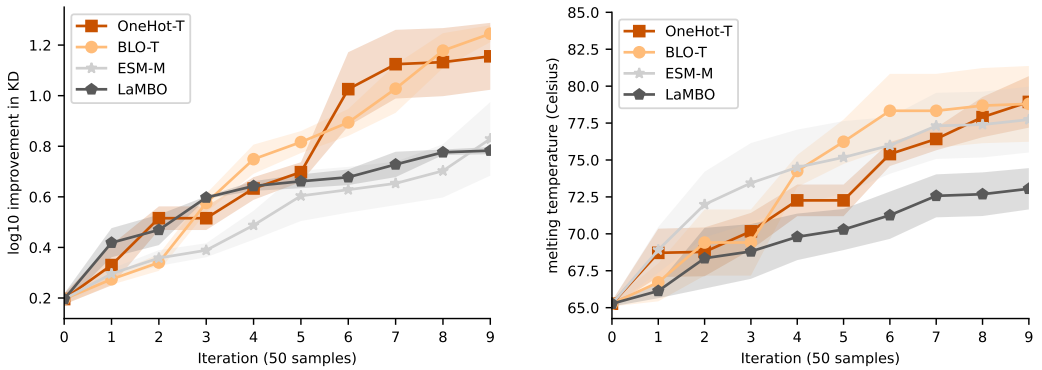
4.2 Structure-based methods

We now consider how incorporating structural information affects the optimization of these properties. We compare the Kermut-T, IgFold-M, IgFold-ESM-M, and IgFold-BLO-T models in Fig. 2. For reference, we also include the overall best sequence-only model, BLO-T. We first observe that none of these methods are able to outperform the sequence-only BLO-T approach for affinity, although IgFold-M performs well in the initial iterations. We hypothesize that this is due to the ability of IgFold-M to more accurately preserve the starting structure, which we explore further in App. E. We also see that IgFold-BLO-T, which combines the IgFold-M structure kernel with the BLO-T sequence kernel, results in a middle ground with decent performance in initial iterations and a peak performance similar to BLO-T.

Finally, Kermut-T far outperforms the other methods, including the sequence-only BLO-T model, when it comes to thermostability. By contrast, its performance on affinity is the worst of the methods

[1] Indeed, we have observed in our in-house experiments that pure-GP BO *does* explore mutations that cause e.g., expression failures.

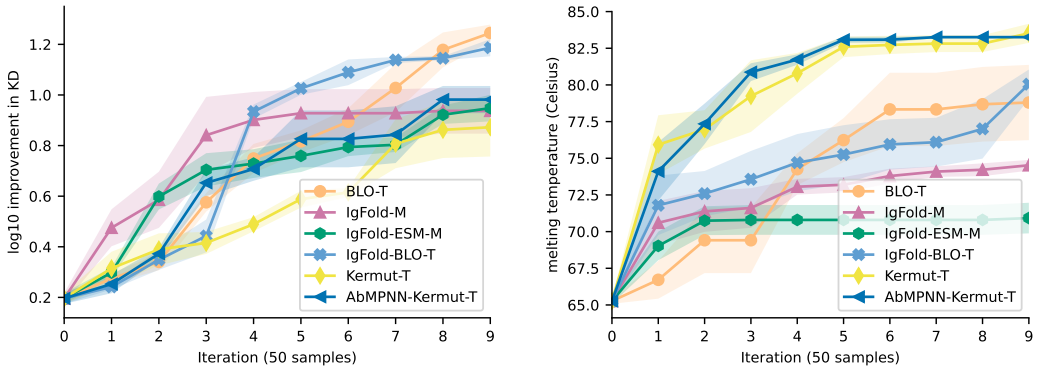
considered here. This points to a fundamental difference in the both features necessary for affinity versus thermostability optimization and the approaches for incorporating structural information.



(a) Optimizing for affinity

(b) Optimizing for T_m

Figure 1: Results on binding affinity K_D and stability T_m for sequence-only approaches. We plot the $\log_{10}$-fold improvement in K_D over the parental sequence for affinity, and the T_m in °C for thermostability.



(a) Optimizing for affinity

(b) Optimizing for T_m

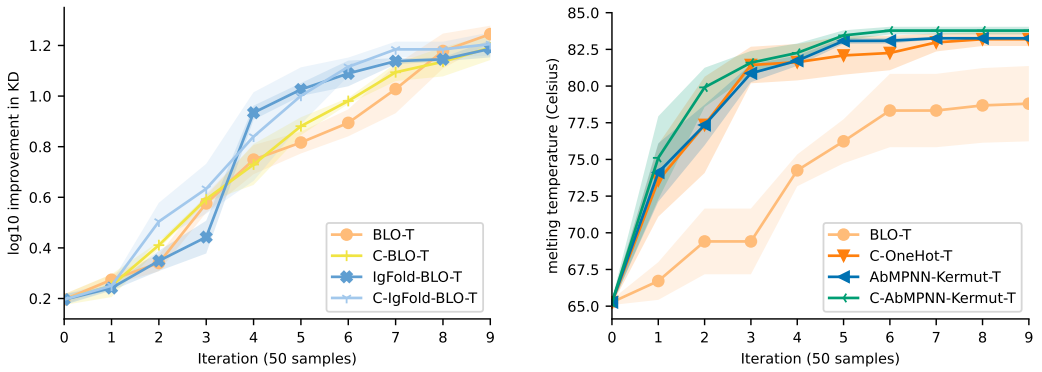
Figure 2: Results on binding affinity K_D and T_m for approaches that incorporate structural information. We also include sequence-only BLO-T for reference.

4.2.1 Does incorporating antibody-specific information help?

In Fig. 2 we also include comparisons to AbMPNN-Kermut-T, which replaces the ProteinMPNN predictions typically used with Kermut with antibody-specific AbMPNN predictions. We observe that while there is little change in the performance on thermostability, its performance on affinity is boosted by the change, particularly in its earlier iterations. These results point to the utility of modality-specific changes. We consider additional modifications to Kermut in App. C.1.

4.3 Do soft constraints help?

We now investigate the effect of incorporating the pLM-based soft constraint of Eq. 1. For clarity, we only show the best-performing method for each setting (sequence-only, structure-based, with and without the pLM soft constraint) in Fig. 3; for full results we refer the reader to App. D. For each method, we use the prefix "C-" to denote the constrained version. We see very little change in affinity optimization from introducing the soft constraint (indeed, the full results show that it can occasionally be detrimental). However, we see that the sequence-based C-OneHot-T is now able to match the performance of the structure-based methods in optimizing thermostability. This result



(a) Optimizing for affinity

(b) Optimizing for T_m

Figure 3: Results on binding affinity K_D and T_m with the inclusion of a pLM-based soft constraint for the best-performing sequence-only and structure-based methods in each setting.

dilutes the advantage these structure-based BO methods have to offer over sequence-based methods in this low-data setting, and eliminates it for thermostability.

5 Discussion & future work

In this work, we have investigated the utility of structure-based BO methods versus sequence-only BO methods in the context of both antibody-intrinsic properties such as developability and antibody-target specific properties such as binding affinity. In doing so, we have explored various means of incorporating sequence and structure deep learning and generative modeling approaches into Bayesian optimization. This led us to assess the impact of different sequence representations, as well as different forms of structural information: "purely structural," working from 3D coordinates alone (e.g., IgFold-M), and "statistical," estimating likelihoods of structural motifs based on a training set of known proteins (e.g., the ProteinMPNN component of Kermut).

We see that for intrinsic properties such as stability, it is important to have a prior which allows comparison to known proteins, but that this can be either purely sequence based (C-OneHot-T) or of the "statistical structural" form (Kermut-T). In this case it does not appear to matter whether this information is antibody-specific.

In contrast, for antibody-target pair specific properties, i.e., binding affinity, peak asymptotic performance requires only sequence information, with some benefit from a domain-specific representation (BLO-T is beneficial, though we do not see any benefits from the sequence representations derived from ESM). Data efficiency in early iterations is aided by "purely structural" information (IgFold-M), which serves primarily to minimize perturbations to the structure of the starting molecule (App. E). Combining the domain-specific sequence representation with this purely structural information provides a compromise in data efficiency and asymptotic performance (IgFold-BLO-T).

However, while some methods do well on both affinity and thermostability, no single method is superior on both, indicating that different features are useful in each case and therefore that an inherent tradeoff exists. One limitation of our work is that the structural information did not include the target structure in the form of an antibody-antigen complex, as this was not available (as is often the case). Addressing this in future work might enable structure-based methods to improve their affinity optimization by encouraging mutations that are more likely to perform well for affinity. Beyond this, the means of incorporating structural information that we considered here were fairly simple: an exciting avenue for future work would be to investigate more sophisticated methods of doing so. Finally, in future work we plan to validate these methods *in vitro*, and to evaluate whether these observations carry over to other developability properties.

Acknowledgments

We would like to thank Nick Bhattacharya and the rest of the BigHat Biosciences DS/ML team for productive discussions and insightful suggestions.

References

[1] Paul J Carter and Greg A Lazar. Next generation antibody drugs: pursuit of the 'high-hanging fruit'. *Nature Reviews Drug Discovery*, 17(3):197–223, 2018.

[2] Alexander Jarasch, Hans Koll, Joerg T Regula, Martin Bader, Apollon Papadimitriou, and Hubert Kettenberger. Developability assessment during the selection of novel therapeutic antibodies. *Journal of pharmaceutical sciences*, 104(6):1885–1898, 2015.

[3] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

[4] Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, Déjenaé L See, Connor Weidle, Riti Biswas, Ellen L Shrock, Philip JY Leung, Buwei Huang, et al. Atomically accurate de novo design of single-domain antibodies. *bioRxiv: The Preprint Server for Biology*, 2024.

[5] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[6] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2022.

[7] Nate Gruver, Samuel Stanton, Nathan Frey, {Tim G.J.} Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and {Andrew Gordon} Wilson. Protein design with guided discrete diffusion. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[8] Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Yucen Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. In *International Conference on Learning Representations (ICLR)*, 2025.

[9] Alexandra Gessner, Sebastian W Ober, Owen Vickery, Dino Oglić, and Talip Uçar. Active learning for affinity prediction of antibodies. *arXiv preprint arXiv:2406.07263*, 2024.

[10] Aniruddh Raghu, Sebastian W Ober, Maxwell Kazman, and Hunter Elliott. Guided sequence-structure generative modeling for iterative antibody optimization. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.

[11] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

[12] Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications*, 14(1):2389, 2023.

[13] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

[14] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[15] David Prihoda, Jad Maamary, Andrew Waight, Veronica Juan, Laurence Fayadat-Dilman, Daniel Svozil, and Danny A. Bitton. BioPhi: A platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning. *mAbs*, 14(1):2020203, 2022.

[16] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[17] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 1978.

[18] Jacob Gardner, Matt Kusner, Xu Zhixiang, Kilian Weinberger, and John Cunningham. Bayesian optimization with inequality constraints. In *International Conference on Machine Learning (ICML)*, 2014.

[19] Mickael Binois, Nicholson Collier, and Jonathan Ozik. A portfolio approach to massively parallel Bayesian optimization. *Journal of Artificial Intelligence Research*, 2025.

[20] Henry Moss, David Leslie, Daniel Beck, Javier Gonzalez, and Paul Rayson. Boss: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 2020.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[22] Ryan-Rhys Griffiths, Leo Klarner, Henry Moss, Aditya Ravuri, Sang Truong, Yuanqi Du, Samuel Stanton, Gary Tom, Bojana Rankovic, Arian Jamasb, et al. GAUCHE: a library for Gaussian processes in chemistry. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Austin Tripp, Sergio Bacallado, Sukriti Singh, and José Miguel Hernández-Lobato. Tanimoto random features for scalable molecular machine learning. In *Advances in Neural Information Processing Systems*, 2023.

[24] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[25] Peter Mørch Groth, Mads Herbert Kerrn, Lars Olsen, Jesper Salomon, and Wouter Boomsma. Kermut: Composite kernel regression for protein variant effects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[26] Frédéric A Dreyer, Daniel Cutting, Constantin Schneider, Henry Kenlay, and Charlotte M Deane. Inverse folding for antibody sequence design using deep learning. *arXiv preprint arXiv:2310.19513*, 2023.

[27] Nate Gruver, Samuel Stanton, Polina Kirichenko, Marc Finzi, Phillip Maffettone, Vivek Myers, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Effective surrogate models for protein design with bayesian optimization. In *ICML Workshop on Computational Biology*, 2021.

[28] Carolin Benjamins, Shikha Surana, Oliver Bent, Marius Lindauer, and Paul Duckworth. Bayesian optimisation for protein sequence design: Gaussian processes with zero-shot protein language model prior mean. In *Machine Learning in Structural Biology Workshop at NeurIPS*, volume 2024, 2024.

[29] Henry Moss, Sebastian W Ober, and Tom Diethe. Return of the latent space COWBOYS: Re-thinking the use of VAEs for Bayesian optimisation of structured spaces. In *International Conference on Machine Learning (ICML)*, 2025.

[30] Taeyoung Yun, Kiyoung Om, Jaewoo Lee, Sujin Yun, and Jinkyoo Park. Posterior inference with diffusion models for high-dimensional black-box optimization. In *International Conference on Machine Learning (ICML)*, 2025.

[31] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

[32] Kevin K Yang, Nicolo Fusi, and Alex X Lu. Convolutions are competitive with transformers for protein sequence pretraining. *Cell Systems*, 15(3):286–294, 2024.

[33] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

[34] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, (ICLR)*, 2015.

[35] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

A Related work

BO for proteins has received an increasing amount of attention in recent years. [20] developed kernels for protein BO and proposed using genetic algorithms for acquisition function optimization. [27] compare different surrogate models for protein BO. [6, 7] propose LaMBO and LaMBO-2, which attempt to perform multi-objective BO by navigating the latent spaces of generative models, with the latter being evaluated on antibody yield and affinity properties. [28] propose incorporating pLM zero-shot predictions into the prior mean of a GP model for protein BO. Similar to our work, [9] evaluate different surrogate models for BO for antibody affinity; however, their evaluation is limited to sequence-only models, affinity, and single-variant acquisitions, limiting their wider applicability. More recently, [8] perform BO using a pLM trained on antibody *families*. Finally, our pLM soft constraint is similar in formulation to recent works combining generative modeling and Bayesian optimization [29, 30]; however, these works focus on modifying generative modeling sampling to perform BO instead of modifying a standard acquisition function.

B Additional methods details

In this section, we give mathematical descriptions of certain methods, where we believe it helpful.

B.1 Portfolio-based acquisition

We briefly describe the qHSRI acquisition function of [19] that we use for batch acquisition. Note that we adjust the equations to be suitable for objective function maximization, which better suits our task. Suppose we have a set of l potential candidates $\mathbf{x}^i$ for the batch, each with predicted mean-standard deviation $\mathbf{a}^i = \left(a_1^i, a_2^i\right) = \left(m\left(\mathbf{x}_i\right), s\left(\mathbf{x}_i\right)\right)$.[2] Then, we define

$$
\begin{aligned}
r_i &= p_{ii}, \\
Q_{ij} &= p_{ij} - p_{ii}p_{jj}, \\
p_{ij} &= \left(\prod_{1\le t\le 2}\left(\min\left(a_t^i, a_t^j\right) - R_t\right)\right) / \left(\prod_{1\le t\le 2}\left(f_t^* - R_t\right)\right),
\end{aligned}
$$

where f_t^* is the maximum observed value for the dimension in the set of candidates, and R is a lower reference point. Then, the "portfolio allocation" is given by $\mathbf{z}^* \in [0,1]^l$ where

$$
\mathbf{z}^* = \underset{\mathbf{z}\in[0,1]^l}{\arg\max}\ h\left(\mathbf{z}\right) = \frac{\mathbf{r}^\top \mathbf{z}}{\sqrt{\mathbf{z}^\top \mathbf{Q}\mathbf{z}}} \quad \text{s.t.} \quad \sum_{i=1}^{l} z_i = 1.
$$

The final batch is given by selecting the candidates with the q highest z_i values. This acquisition function favors candidates on the predicted mean-standard deviation Pareto front. Intuitively, this approach can be seen as a method for selecting different values of the β parameter in the upper confidence bound acquisition function [31]:

$$
a_{\text{UCB}}\left(\mathbf{x}\right) = m\left(\mathbf{x}\right) + \beta\sigma\left(\mathbf{x}\right),
$$

where $\mathbf{m}\left(\cdot\right)$ and $\sigma\left(\cdot\right)$ are the posterior mean and standard deviation, and forming a batch from optimizing the resulting acquisition functions separately.

In order to modify this with our pLM soft constraint, we multiply the resulting r_i values by the pLM likelihoods. Note that we modify the genetic algorithm to optimize for pLM probabilities alongside the predicted mean and standard deviations.

B.2 Kermut

We briefly describe the salient features of the Kermut model [25] for our analysis. The Kermut model is a GP model with a particular choice of kernel and mean function. The kernel is made up of a structure component and a sequence component in a weighted sum:

$$
k\left(\mathbf{x}, \mathbf{x}'\right) = \pi k_{\text{struct}}\left(\mathbf{x}, \mathbf{x}'\right) + (1-\pi)k_{\text{seq}}\left(\mathbf{x}, \mathbf{x}'\right).
$$

[2] Recall that we find the candidates by running a genetic algorithm to find the (approximate) predicted mean-standard deviation Pareto front.

The sequence kernel is chosen to be a squared exponential kernel on the mean-pooled embeddings from ESM-2. The structure kernel involves three parts, summed over the effect from each residue that differs from the parental sequence:

$$k_{\text{struct}}(\mathbf{x}, \mathbf{x}') = \sum_{i \in M} \sum_{j \in M'} k^1_{\text{struct}}\left(\mathbf{x}_i, \mathbf{x}'_j\right),$$

where M and M' are the sets of mutated residues (with respect to the parental). k^1_{struct} itself is made up of three separate kernels:

$$k^1_{\text{struct}}(\mathbf{x}, \mathbf{x}') = \lambda k_H(\mathbf{x}, \mathbf{x}') k_p(\mathbf{x}, \mathbf{x}') k_d(\mathbf{x}, \mathbf{x}').$$

Here, $\lambda > 0$ is a scalar, k_H represents a Hellinger distance-based kernel on probabilities from an inverse folding model, k_p represents an exponential kernel on the inverse folding probabilities, and k_d is a kernel acting on the physical distance between residues.

The final component of Kermut is the prior mean function, which is chosen to be

$$m(\mathbf{x}) = \alpha f_0(\mathbf{x}) + \beta.$$

In this case, $f_0(\cdot)$ is chosen to be an ESM-2 zero-shot log-likelihood ratio between variant and wild-type sequences:

$$f_0(\mathbf{x}) = \sum_{i \in M} \log \mathrm{p}(\mathbf{x}_i) - \log \mathrm{p}\left(\mathbf{x}_i^{\text{WT}}\right).$$

Note that for multiple mutations, the sum of each individual mutation is taken independently, instead of re-calculating the (pseudo-)log likelihood based on all mutations.

B.3 Affinity and thermostability oracles

The oracles we used were derived from ensembles of 10 CARP/ByteNet regressors [32]. The affinity ensemble was pretrained on approximately 285,000 sequences from phage display, processed using Next Generation Sequencing (NGS), and fine-tuned on 6,881 sequences with K_D data obtained from Bio-Layer Interferometry (BLI). Our thermostability ensemble was pretrained on approximately 537,000 sequences from NGS phage display, and 9556 T_m datapoints obtained from NanoDSF. The affinity ensemble achieved a test cross-validated Spearman correlation of 0.95, whereas the thermostability ensemble achieved a correlation of 0.72. Note that for our evaluation, we only use the first model of the ensemble for computational speed. Finally, we use 159 variants from the early stages of the campaign as the starting set for subsampling for our BO runs.

C An ablation study on Kermut

Here, we briefly compare different versions of Kermut, motivating the modifications that we take forward. The first set of modifications, which we denote as Kermut-M, involves a few minor changes, but attempts to keep the overall model largely unchanged. We first modify the Kermut code to ensure that everything is computed in double precision. We also disable GPyTorch's `fast_computations` settings [33] to ensure exact, Cholesky-based GP inference. We additionally replace the Adam-based training [34] with BoTorch's `fit_gpytorch_mll` method, which uses L-BFGS [35]. Finally, we notice that the parameterization of the final Kermut kernel is effectively

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \pi k_{\text{struct}}(\mathbf{x}, \mathbf{x}') + (1 - \pi) k_{\text{seq}}(\mathbf{x}, \mathbf{x}'),$$

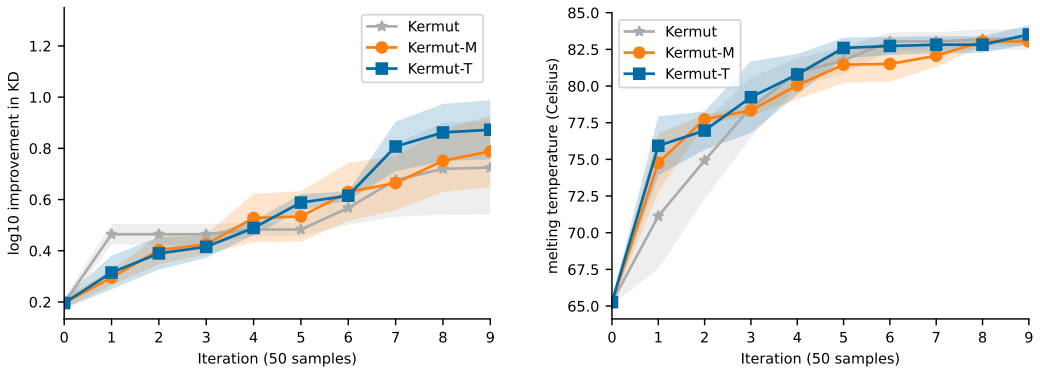
where σ_f^2 is the GP signal variance, $k_{\text{struct}}(\cdot, \cdot')$ is the structure kernel, $k_{\text{seq}}(\cdot, \cdot')$ is the sequence kernel, and $\pi \in (0, 1)$ is a weighting. We hypothesize that a better parameterization is instead

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(\pi k_{\text{struct}}(\mathbf{x}, \mathbf{x}') + (1 - \pi) k_{\text{seq}}(\mathbf{x}, \mathbf{x}')\right),$$

and implement this instead.

Finally, given the relative performance of ESM-based embeddings and one-hot Tanimoto kernels, and given the much greater computational cost of the ESM embeddings, we investigate replacing the default ESM embedding RBF sequence kernel with the one-hot Tanimoto kernel, leading to Kermut-T. Note that we otherwise retain the modifications used in Kermut-M.

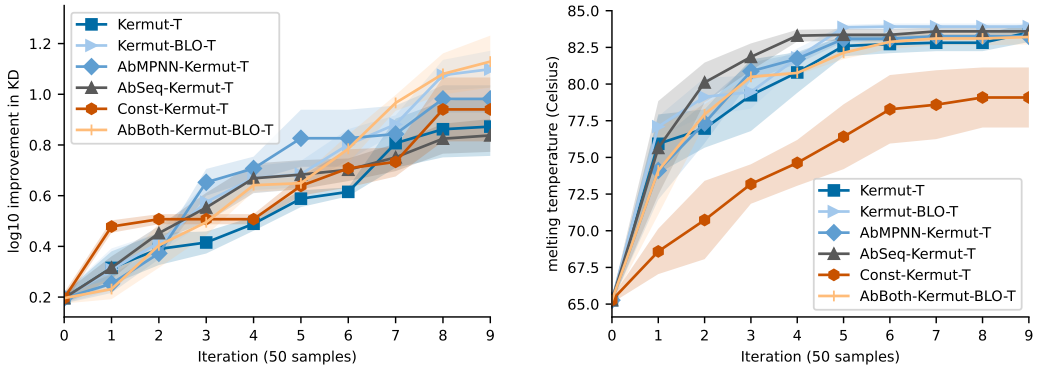
We plot the results of these experiments on our *in silico* evaluation in Fig. 4, showing that Kermut-M and Kermut-T match if not outperform the baseline Kermut, justifying our modifications.



(a) Optimizing for affinity

(b) Optimizing for T_m

Figure 4: Results on binding affinity K_D and T_m for different modifications of Kermut.



(a) Optimizing for affinity

(b) Optimizing for T_m

Figure 5: Results on binding affinity K_D and T_m for antibody-specific modifications of Kermut.

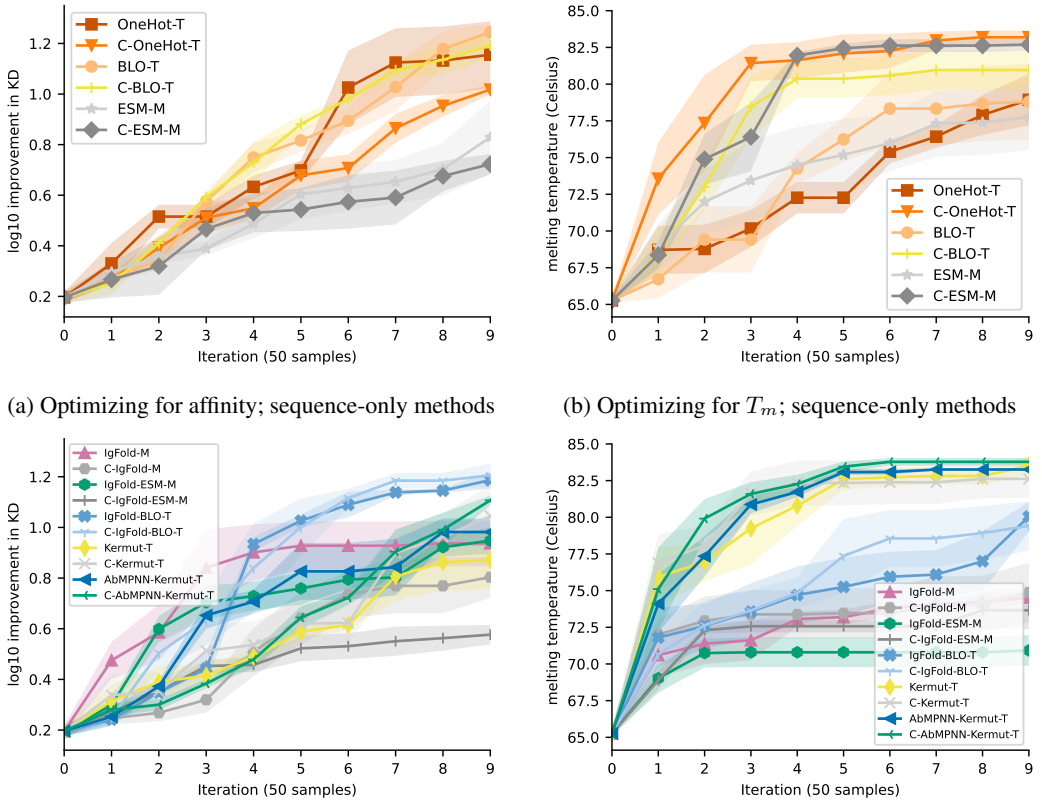
C.1 Further antibody-specific improvements of Kermut

We now try to ablate different modifications that we could make to Kermut, in hopes of improving its performance. We focus first on replacing the protein-related deep models involved in Kermut with antibody-specific versions. To this end, we individually ablate:

- replacing ProteinMPNN with AbMPNN (AbMPNN-Kermut-T);
- replacing ESM-2 in the prior mean with a pLM trained on SAbDab (AbSeq-Kermut-T);
- replacing the prior mean with a simple learned constant (Const-Kermut-T); and,
- given the success of the BLOSUM-based encoding in the sequence-based models, replacing the one hot encoding in the Tanimoto sequence kernel module with the BLOSUM-based encoding (Kermut-BLO-T).

Finally, we attempt combining all the antibody-specific models (AbMPNN and SAbDab-trained MLM) and the BLOSUM encoding into a final variation, AbBoth-Kermut-BLO-T.

We show these results in Fig. 5. These results show that overall, the antibody-specific modifications are helpful. However, Kermut-B seems to perform the best out of all methods, as the combination of all the antibody-based modifications do not seem to be beneficial together. Finally, we observe that the use of *some* form of pLM in the prior mean seems beneficial, particularly for T_m.



(a) Optimizing for affinity; sequence-only methods

(b) Optimizing for T_m; sequence-only methods

(c) Optimizing for affinity; structure-based methods

(d) Optimizing for T_m; structure-based methods

Figure 6: Results on binding affinity K_D and T_m with the inclusion of a pLM-based soft constraint. Note that we separate out sequence-only (top) and structure-based (bottom) methods for clarity.

D Full soft constraint results

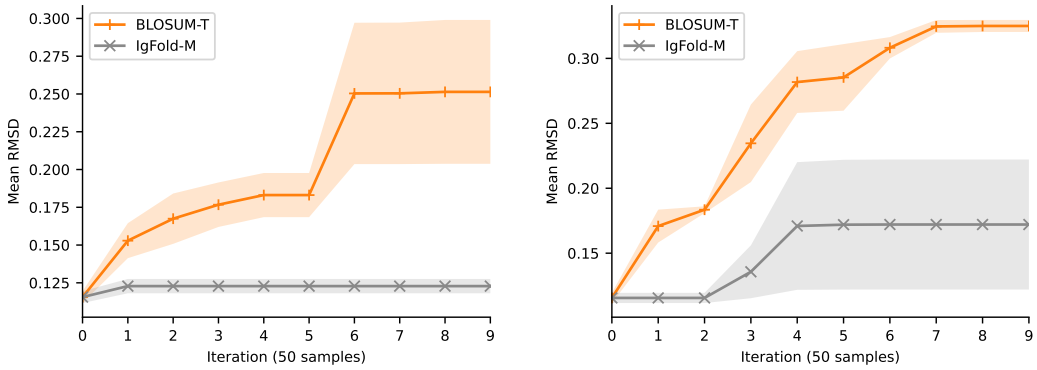
We plot the full results for the soft constraint experiment, separated by sequence-only and structure-based methods, in Fig. 6.

E Structural exploration results

In Figure 7 we plot the (predicted) RMSDs between the parental and the proposals over the course of BO iterations. We see that the sequence-based approach diverges further from the parental in structure-space than the structure-based approach. This indicates that the structure-based method is better able to hone in on the structural conformation that is most promising for the property at hand. In particular, for affinity, we do not expect the conformation to change drastically from the parental antibody's conformation when doing iterative optimization.

F A summary of our methods

In Table 1 we summarize the methods evaluated in our work, describing how each of them utilizes sequence and/or structure information.



(a) Optimizing for affinity

(b) Optimizing for T_m

Figure 7: Measuring structural exploration from parental when optimizing (a) binding affinity K_D and (b) T_m. RMSDs are computed after aligning the IgFold-predicted structures for both the parent and proposed sequences. We compare RMSDs for BLO-T, representing sequence-only optimization, and IgFold-M, representing structure-only optimization.

Method Name	Prior Mean	Seq Rep	Seq Kernel	Struct	Seq-Struct Combo	Constraint
OneHot-T	const	One-hot	Tanimoto	None	NA	None
C-OneHot-T	const	One-hot	Tanimoto	None	NA	Sapiens pl
BLO-T	const	BLOSUM	Tanimoto	None	NA	None
C-BLO-T	const	BLOSUM	Tanimoto	None	NA	Sapiens pl
ESM-M	const	ESM2 emb	Matérn-5/2	None	NA	None
C-ESM-M	const	ESM2 emb	Matérn-5/2	None	NA	Sapiens pl
IgFold-M	const	None	NA	IgFold coords	None	None
C-IgFold-M	const	None	NA	IgFold coords	None	Sapiens pl
IgFold-ESM-M	const	ESM2 emb	None	IgFold coords	Concat, Matérn-5/2	None
C-IgFold-ESM-M	const	ESM2 emb	None	IgFold coords	Concat, Matérn-5/2	Sapiens pl
IgFold-BLO-T	const	BLOSUM	Tanimoto	IgFold coords	Add kernels	None
C-IgFold-BLO-T	const	BLOSUM	Tanimoto	IgFold coords	Add kernels	Sapiens pl
Kermut-T	ESM2 pll	One-hot	Tanimoto	Composite	Add kernels	None
C-Kermut-T	ESM2 pll	One-hot	Tanimoto	Composite	Add kernels	Sapiens pl
AbMPNN-Kermut-T	ESM2 pll	One-hot	Tanimoto	Composite (Ab)	Add kernels	None
C-AbMPNN-Kermut-T	ESM2 pll	One-hot	Tanimoto	Composite (Ab)	Add kernels	Sapiens pl
Const-Kermut-T	const	One-hot	Tanimoto	Composite	Add kernels	None
AbSeq-Kermut-T	SAbDab pll	One-hot	Tanimoto	Composite	Add kernels	None
Kermut-BLO-T	ESM2 pll	BLOSUM	Tanimoto	Composite	Add kernels	None
AbBoth-Kermut-BLO-T	SAbDab pll	BLOSUM	Tanimoto	Composite (Ab)	Add kernels	None

Table 1: Summary of methods evaluated in this work.