Ab-L3BO: Bayesian Optimization of Antibody Sequence Design with Large Language Models

Eunna Huh[1] **Seungjin Choi**[2] **Hyunjin Shin**[1,†]

[1]MOGAM Institute for Biomedical Research, Korea
[2]CROID Research and aSSIST University, Korea
[†]Correspondence to hyunjin.shin@mogam.re.kr

Abstract

Enhancing the properties of therapeutic antibodies requires modifying amino acid sequences, typically through iterative engineering and experimental validation such as binding assays, which are time- and resource-intensive. To address these challenges, we introduce Ab-L3BO, a Bayesian optimization (BO) framework that integrates an antibody-specific large language model (LLM) with Bayesian fine-tuning. The Ab-L3BO leverages prior knowledge embedded in an antibody-specific LLM and efficiently updates its parameters using a Bayesian Low-Rank Adaptation (LoRA) with Laplace approximation (LA). The fine-tuned LLM serves as a surrogate model within the BO framework. Our results show that Ab-L3BO achieves superior sample efficiency than conventional BO methods based on Gaussian process regression. Integrating LLMs into the BO framework offers three key advantages: (1) rapid adaptation to new domains; (2) warm-starting optimization with pretrained knowledge; (3) bypassing the complexities of kernel design choices, kernel computation, and sequence alignment to handle variable-length sequences. Overall, our study demonstrates that incorporating LLMs into BO substantially improves the efficiency of antibody sequence design, potentially reducing both time and resource requirements in therapeutic antibody development.

1. Introduction

Recent market analysis (Grand View Research (2023)) indicates rapid growth of the monoclonal antibody (mAb) market, highlighting its importance in modern therapeutics. mAbs are designed to mimic natural immune responses by selectively binding disease-associated antigens involved in cancers, autoimmune disorders, and infectious diseases (Jin et al. (2022); Puthenpurail et al. (2021); Hafeez et al. (2018)). Despite advances in in vitro screening, predicting the relationship between antibody sequence (genotype) and binding affinity (phenotype) remains a major challenge. In practice, this relationship is typically explored through empirical strategies involving iterative sequence engineering followed by experimental validation until the desired phenotype is achieved, which are time- and resource-intensive. Computational methods provide a promising alternative by systematically exploring the antibody sequence–function landscape, improving the efficiency of therapeutic antibody development.

Given that antibody sequences are categorial, with each position taking on one of 20 amino acids, specialized kernel functions are required to model such discrete structures. However, kernel materialization—explicitly computing and storing the kernel matrix in GP surrogate models—can impose substantial computational burden. Selecting appropriate kernels and fine-tuning their hyperparameters is particularly challenging for high-dimensional categorial sequence data. Additionally, the high

variability in both amino acid composition and sequence length among antibodies introduce further complications. Consequently, preprocessing steps such as sequence alignment are often employed, although these procedures are computationally intensive and prone to alignment errors caused by gap insertions.

Our contribution:

We introduce Antibody-Laplace-LoRA-LLM Bayesian Optimization (Ab-L3BO) inspired by Kristiadi et al. (2024), an approach that integrates a LLM with BO for the purpose of optimizing antibody sequences toward enhanced binding affinity. Building on the representations learned from antibody-specific LLMs, Ab-RoBERTa (Huh et al. (2025)), we utilized it as a surrogate model within BO framework, incorporating Low-Rank Adaptation (LoRA) (Hu et al. (2021)) and Laplace approximation (LA) (Daxberger et al. (2022)).

(1) **Flexible optimization of biological sequences.** By leveraging LLM-based representations from Ab-RoBERTa within BO, our approach bypasses the need for manual kernel design, kernel computation, and sequence alignment, allowing efficient optimization of variable-length antibody sequences.

(2) **Parameter-efficient adaptation with LoRA and local search.** We jointly train a regression head on the last layer of Ab-RoBERTa while applying LoRA—a parameter-efficient fine-tuning (PEFT). This allows warm-starting the optimization using pretrained knowledge. As a result, it substantially reduces computational and memory costs and lowers data requirements while maintaining performance comparable to full fine-tuning. Additionally, following prior work on local probabilistic search for BO (Wan et al. (2021)), we explore candidate sequences within the k-nearest neighbors of the current best sequence in embedding space, achieving a 3.6× lower K_D than random candidate selection.

(3) **Uncertainty quantification with Laplace approximation.** In parallel, LA enables efficient uncertainty quantification by approximating the posterior distribution of the Ab-RoBERTa parameters with a Gaussian centered at the maximum a posteriori (MAP) estimate (Daxberger et al. (2022)). This provides efficient uncertainty estimation across the Ab-RoBERTa parameters, thereby enhancing the robustness and calibration of the Ab-RoBERTa utility in BO framework.

2. Method: Bayesian LoRA of LLM Surrogate

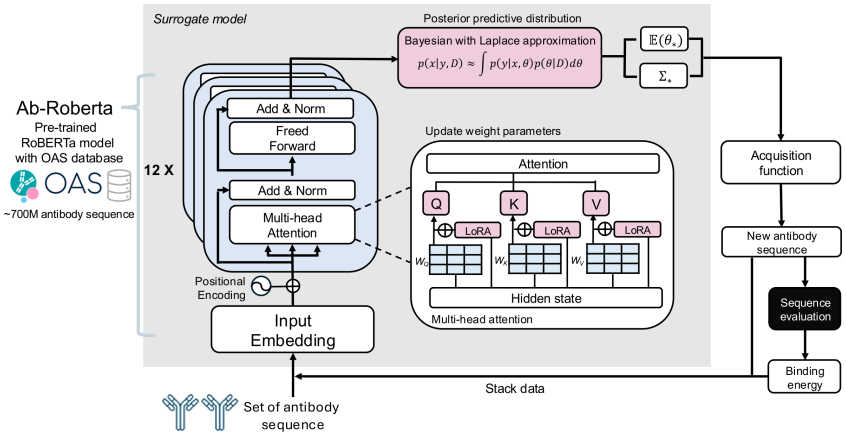


Figure 1: Overview of Ab-L3BO architecture.

Antibody evolutionary fitness is driven by selective pressures for antigen recognition, making biologically informative embeddings essential. To identify an appropriate representation model, we compared five pre-trained LLMs — ProtT5, ProtBERT (Elnaggar et al. (2022)), IgT5, IgBERT (Kenlay et al. (2024)), and Ab-RoBERTa (Huh et al. (2025)) — by encoding 2000 randomly selected antibody heavy-chain sequences and visualizing the embeddings with UMAP. Antibody-specific models (IgT5, IgBERT, Ab-RoBERTa) distinguished B-cell subtypes and germline V genes, whereas general protein models (ProtT5, ProtBERT) separated only germline V genes. Considering parameter size and update efficiency, Ab-RoBERTa was selected as the surrogate model for the BO framework (Appendix G).

To integrate Ab-RoBERTa into the BO framework, we adopted a Bayesian fine-tuning strategy combining LoRA (Hu et al. (2021)) and LA (Daxberger et al. (2022)), following the approach described by Kristiadi et al. (2024). The pretrained Ab-RoBERTa backbone remains fixed while a regression head and LoRA parameters adapt the model to the antibody–affinity prediction task. LoRA introduces trainable low-rank matrices into the self-attention layers, enabling parameter-efficient updates. The probability distribution of Ab-RoBERTa parameters given data is approximated using LA, enabling Bayesian fine-tuning of the surrogate model within the BO loop.

At each BO iteration, Ab-RoBERTa encodes antibody sequences into contextual embeddings by averaging the last hidden state while excluding padding and EOS tokens. These embeddings are used to jointly train the LoRA modules and a regression head composed of two fully connected layers with 100 hidden units each. Training uses learning rates of 3e-4 (LoRA) and 1e-3 (regression head) with 1e-2 weight decay for 500 epochs, followed by an additional 200 epochs of regression-head fine-tuning. LoRA was configured with rank $r = 16$ and $\alpha = 8$; parameters are Gaussian-initialized with a dropout rate of 15% (Appendix Table 1).

For uncertainty estimation, LA was applied to the parameters of both LoRA and regression head using the `laplace-bayesopt` package (Kristiadi et al. (2023)). Given a MAP estimate θ_*, the posterior distribution $q(\theta \mid D)$ is approximated as a Gaussian $\mathcal{N}(\theta_*, \Sigma_*)$, where the covariance matrix Σ_* is calculated as the inverse of Hassian of the negative log-likelihood θ_*:

$$\Sigma_* = \left(-\nabla^2_\theta \log p(\theta \mid D)\big|_{\theta=\theta_*}\right)^{-1}.$$

The Hessian was approximated using a post hoc Kronecker-Factored Approximate Curvature (KFAC) (Heskes (2000); Martens and Grosse (2020); Botev et al. (2017)), with a low-rank eigendecomposition for computational efficiency Daxberger et al. (2022). This posterior approximation provides uncertainty estimates for the surrogate model, enabling principled acquisition optimization in the BO loop (Appendix A). The overall workflow is summarized in Algorithm 1.

Algorithm 1 Ab-L3BO

Require: Objective function f, pre-trained LLM model $\hat{f}$, feasible set of antibody sequences $D_{\text{cand}} = \{x_i\}_{i=1}^n$, initial dataset $D_1 = \{(x_i, f(x_i))\}_{i=1}^m$, number of evaluations T, number of nearest neighbors κ

1: **for** $t = 1, \ldots, T$ **do**
2: Fine-tune the LLM surrogate model $\hat{f}_t$ by LA-LoRA using dataset D_t
3: Construct $KNN_t \subseteq D_{\text{cand}}$, which contains κ nearest neighbors around the current best point $x^* = \arg\min_{x \in D_t} \hat{f}_t(x)$
4: Optimize the acquisition function over KNN_t: $x_t = \arg\min_{x \in KNN_t} \alpha(x \mid D_t)$
5: Evaluate the reward $f(x_t)$
6: Augment the data: $D_{t+1} = D_t \cup \{(x_t, f(x_t))\}$
7: Update the feasible set: $D_{\text{cand}} = D_{\text{cand}} \setminus \{x_t\}$
8: **end for**
9: **return** $\arg \min_{(x, f(x)) \in D_{T+1}} f(x)$

3. Experiments

3.1. Datasets

The AAYL49 dataset (Engelhart et al. (2022)) contains single-chain variable fragment (scFv) antibody sequences targeting a SARS-CoV-2 peptide (PDVDLGDISGINAS), with binding affinity measured using the AlphaSeq assay. In total, 19,401 unique heavy-chain sequences and their corresponding affinity measurements were collected.

The Absolut! dataset contains 999,985 unique CDRH3 sequences targeting the IgG4 Fc region (PDB: 1ADQ), with binding affinity estimation using the Absolut! simulator (Robert et al. (2022)). Binding conformations are determined by analyzing 11-mer subsequences from antibody and antigen sequences. As a result, 839,777 unique 11-mer CDRH3 sequences were extracted, representing energetically optimal binding segments for the target antigen.

3.2. Impact of Local Search and Initial Dataset Size on Ab-L3BO Optimization

Previous studies have shown that BO performance can be enhanced through local probabilistic search strategies for global optimization (Khan et al. (2023); Wan et al. (2021); Kristiadi et al. (2024); Eriksson et al. (2020)). Building on this concept, we conducted a local search in the embedding space by sampling candidate sequences within the k-nearest neighbors ($k = 5000$) of the current best sequence using the AAYL49 antibody dataset. This strategy substantially improved optimization outcomes: the lowest K_D obtained through local search was 3.6-fold lower than that achieved by random candidate selection (t-test p-value $= 0.0048$), demonstrating the effectiveness of embedding-based local search (Appendix F).

Within this framework, we next investigated how the size of the initial dataset influences optimization performance. As expected, performance improved as the number of initial sequences increased, with 500 initial sequences yielding statistically significant gains. However, such dataset sizes may be impractical in wet-lab experimental settings. While the optimal initialization size may vary across datasets, our results suggest that starting with 50 initial sequences provides a practical compromise, as the mean lowest K_D across five iterations remained in the picomolar range with minimal variation. (Appendix H).

3.3. Kernel Materialization Complexity Avoided by Ab-L3BO

Antibody sequences consist of 20 amino acids, and previous studies modeled them as categorical variables using one-hot encoding with kernel functions for Gaussian Process (GP) modeling (Khan et al. (2023); Deshwal et al. (2023); Moss et al. (2020)). Meanwhile, Ab-RoBERTa converts sequences into a continuous representation and serves directly as a surrogate model, capturing contextual sequence information while avoiding explicit kernel computation.

To evaluate sampling and computational efficiency, we compared Ab-L3BO with three kernel-based BO methods: transformed overlap kernel (TOK) (Khan et al. (2023); Wan et al. (2021)), Hamming embedding via dictionaries (HED) (Deshwal et al. (2023)), and the sub-sequence string kernel (SSK) (Moss et al. (2020); Cancedda et al. (2003)) (Appendix C). Ab-L3BO consistently outperformed all kernel-based methods and random search, likely benefiting from the warm-start initialization provided by the pretrained Ab-RoBERTa model (Figure 2).

In terms of computational efficiency, we observed that acquisition function optimization accounted for the majority of Ab-L3BO's runtime, with the time cost increasing as the number of candidate samples increased. This observation is consistent with findings from a previous study (Kristiadi et al. (2024)). Among the compared methods, SSK incurred the highest computational cost (71,819 seconds $\approx$ 19 hours), followed by Ab-RoBERTa, TOK, and HED, which required 192, 0.366, and 0.053 seconds, respectively (Figure 2).

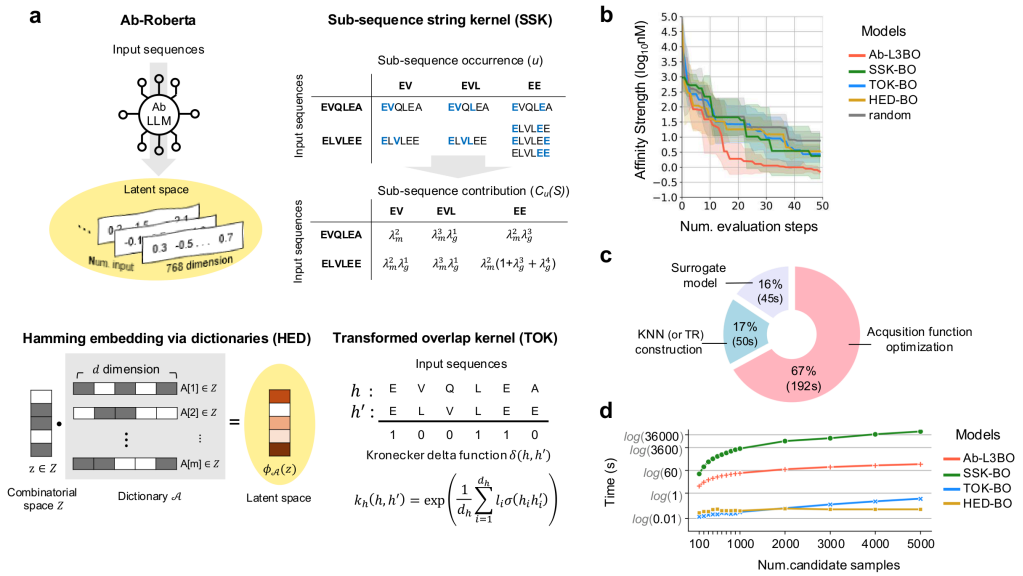


Figure 2: **The performance comparison using different sequence representation methods. (a)**, Four different sequence representation methods are illustrated. **(b)**, BO performance is compared across five representations, averaged over five runs. **(c)**, Runtime composition (seconds) for a single Ab-L3BO step. **(d)**, Acquisition optimization time across four representations as the candidate set size increases.

3.4. Generalization of Ab-L3BO Across Independent Antibody Datasets

To evaluate generalization, we applied Ab-L3BO to an independent dataset from the Absolut! database (Robert et al. (2022)). Binding affinities were estimated using the Absolut! simulator based on the Miyazawa–Jernigan potential (Miyazawa and Jernigan (1996)), producing both variable-length CDRH3 and fixed 11-mer datasets. Across five runs, Ab-L3BO achieved the best mean optimization performance compared with four baseline models (Appendix I). Together with the results on AAYL49 (Figure 2), these findings indicate that Ab-L3BO generalizes across heterogeneous datasets while maintaining robust optimization performance.

4. Conclusion

Ab-L3BO integrates an antibody-specific LLM with Bayesian optimization to enable efficient antibody sequence design. By leveraging pretrained representations from Ab-RoBERTa, the framework benefits from warm-start initialization, avoids explicit kernel construction, and directly handles variable-length sequences. Evaluations on two distinct datasets, AAYL49 and Absolut!, demonstrate that Ab-L3BO generalizes across heterogeneous data sources, consistently achieving superior optimization performance while maintaining competitive computational efficiency (Appendix I).

Currently, Ab-L3BO optimizes over a finite candidate pool, which reduces computational cost but restricts exploration to predefined sequences. Extending the framework to larger search spaces is a promising direction for future work.

References

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization, 2020. URL https://arxiv.org/abs/1910.06403.

Daniel Beck and Trevor Cohn. Learning kernels over strings using gaussian processes. In Greg Kondrak and Taro Watanabe, editors, *Proceedings of the Asian Federation of Natural Language Processing (AFNLP)*, Taipei, Taiwan, 2017. Asian Federation of Natural Language Processing.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning, 2017. URL https://arxiv.org/abs/1706.03662.

N. Cancedda, E. Gaussier, C. Goutte, and J. M. Renders. Word-sequence kernels. *Journal of Machine Learning Research*, pages 1059–1082, 2003.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux – effortless bayesian deep learning, 2022. URL https://arxiv.org/abs/2106.14806.

Aryan Deshwal, Sebastian Ament, Maximilian Balandat, Eytan Bakshy, Janardhan Rao Doppa, and David Eriksson. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings, 2023. URL https://arxiv.org/abs/2303.01774.

A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

E. Engelhart, R. Emerson, L. Shing, C. Lennartz, D. Guion, M. Kelley, C. Lin, R. Lopez, D. Younger, and M. E. Walsh. A dataset comprised of binding interactions for 104,972 antibodies against a sars-cov-2 peptide. *Scientific Data*, 9(1):653, 2022. doi: 10.1038/s41597-022-01779-4.

David Eriksson, Michael Pearce, Jacob R Gardner, Ryan Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization, 2020. URL https://arxiv.org/abs/1910.01739.

Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration, 2021. URL https://arxiv.org/abs/1809.11165.

Grand View Research. Monoclonal antibodies market size & share report, 2030, 2023. URL https://www.grandviewresearch.com/industry-analysis/monoclonal-antibodies-market.

Ryan-Rhys Griffiths, Leo Klarner, Henry B. Moss, Aditya Ravuri, Sang Truong, Samuel Stanton, Gary Tom, Bojana Rankovic, Yuanqi Du, Arian Jamasb, Aryan Deshwal, Julius Schwartz, Austin Tripp, Gregory Kell, Simon Frieder, Anthony Bourached, Alex Chan, Jacob Moss, Chengzhi Guo, Johannes Durholt, Saudamini Chaurasia, Felix Strieth-Kalthoff, Alpha A. Lee, Bingqing Cheng, Alán Aspuru-Guzik, Philippe Schwaller, and Jian Tang. Gauche: A library for gaussian processes in chemistry, 2023. URL https://arxiv.org/abs/2212.04450.

U. Hafeez, H. K. Gan, and A. M. Scott. Monoclonal antibodies as immunomodulatory therapy against cancer and autoimmune diseases. *Current Opinion in Pharmacology*, 41:114–121, 2018. doi: 10.1016/j.coph.2018.05.010.

Tom Heskes. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 12(4):881–901, 2000. doi: 10.1162/089976600300015637.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Eunna Huh, Hyeonsu Lee, and Hyunjin Shin. Antibody foundational model : Ab-roberta, 2025. URL https://arxiv.org/abs/2506.13006.

S. Jin, Y. Sun, X. Liang, X. Gu, J. Ning, Y. Xu, S. Chen, and L. Pan. Emerging new therapeutic antibody derivatives for cancer treatment. *Signal Transduction and Targeted Therapy*, 7(1):39, 2022. doi: 10.1038/s41392-021-00868-x.

Henry Kenlay, Frédéric A. Dreyer, Aleksandr Kovaltsuk, Dom Miketa, Douglas Pires, and Charlotte M. Deane. Large scale paired antibody language models, 2024. URL https://arxiv.org/abs/2403.17889.

A. Khan, A. I. Cowen-Rivers, A. Grosnit, D. Deik, P. A. Robert, V. Greiff, E. Smorodina, P. Rawat, R. Akbar, K. Dreczkowski, et al. Toward real-world automated antibody design with combinatorial bayesian optimization. *Cell Reports Methods*, 3(1):100374, 2023. doi: 10.1016/j.crmeth.2022.100374.

Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, and Vincent Fortuin. Promises and pitfalls of the linearized laplace in bayesian optimization, 2023. URL https://arxiv.org/abs/2304.08309.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules?, 2024. URL https://arxiv.org/abs/2402.05015.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, pages 419–444, 2002.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature, 2020. URL https://arxiv.org/abs/1503.05671.

S. Miyazawa and R. L. Jernigan. Residue–residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *Journal of Molecular Biology*, 256(3):623–644, 1996. doi: 10.1006/jmbi.1996.0114.

Henry B. Moss, Daniel Beck, Javier Gonzalez, David S. Leslie, and Paul Rayson. Boss: Bayesian optimization over string spaces, 2020. URL https://arxiv.org/abs/2010.00979.

A. Puthenpurail, H. Rathi, S. M. Nauli, and A. Ally. A brief synopsis of monoclonal antibody for the treatment of various groups of diseases. *World Journal of Pharmacy and Pharmaceutical Sciences*, 10(11):14–22, 2021.

P. A. Robert, R. Akbar, R. Frank, M. Pavlović, M. Widrich, I. Snapkov, A. Slabodkin, M. Chernigovskaya, L. Scheffer, E. Smorodina, et al. Unconstrained generation of synthetic antibody–antigen structures to guide machine learning methodology for antibody specificity prediction. *Nature Computational Science*, 2(12):845–865, 2022. doi: 10.1038/s43588-022-00372-4.

Binxin Ru, Ahsan S. Alvi, Vu Nguyen, Michael A. Osborne, and Stephen J Roberts. Bayesian optimisation over multiple continuous and categorical inputs, 2020. URL https://arxiv.org/abs/1906.08878.

Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces, 2021. URL https://arxiv.org/abs/2102.07188.

A. Bayesian Optimization (BO)

BO is a sample-efficient method for identifying the global optimum of an expensive-to-evaluate black-box function $f(x)$. In the context of antibody design, BO is used to find an antibody sequence $x \in \mathcal{X}$ that minimizes the binding affinity measured by $f(x)$, where lower values indicate stronger binding, with as few evaluations as possible. Since the function $f(x)$ is unknown, BO builds a surrogate model $\hat{f}$ to approximate f using a set of available data points. A common choice is Gaussian Process (GP) regression, which provides both a predictive mean $\mu(x)$ and an uncertainty estimate $\sigma(x)$ for each input x. Given a small number of initial data points, GP regression calculates the posterior distribution over functions analytically under a GP prior and Gaussian likelihood.

BO selects the next evaluation point by optimizing an acquisition function $a(x)$, which balances exploration and exploitation. Common acquisition functions include Expected Improvement (EI), GP Upper Confidence Bound (GP-UCB), and Thompson Sampling (TS).

For all BO experiments—including those utilizing Ab-L3BO, SSK (Moss et al. (2020); Cancedda et al. (2003); Lodhi et al. (2002)), HED (Deshwal et al. (2023)), and TOK (Wan et al. (2021))—we used TS, where a random acquisition function is directly sampled from the posterior over the function, i.e.,

$$\hat{f} \sim p(\hat{f} \mid D),$$

and the next candidate is selected as

$$x_{\text{next}} = \arg\min_{x \in \mathcal{X}} \hat{f}(x). \tag{1}$$

The function value at the selected point is evaluated, appended to the dataset, and used to update the surrogate model. This process—surrogate modeling followed by acquisition function optimization—is repeated until convergence or until the evaluation budget is exhausted. We implemented BO using the `BoTorch` (Balandat et al. (2020)) library (v0.11.3), built on top of `PyTorch`, which provides scalable and flexible implementations of BO algorithms.

B. Gaussian Process (GP) Surrogate

Gaussian Process (GP) regression is a nonparametric Bayesian method for modeling unknown functions and is particularly useful when quantifying uncertainty is important. A GP defines a distribution over functions f, such that any finite set of function values follows a joint Gaussian distribution. Formally, the GP is specified by a mean function $\mu(x)$ and a covariance function (kernel function) $k(x, x')$:

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')). \tag{2}$$

Typically, the mean function is assumed to be zero, i.e., $\mu(x) = 0$, for simplicity, and the kernel function encodes assumptions about the smoothness or similarity of the function.

Given a set of training data

$$D = \{(x_i, y_i)\}_{i=1}^{n},$$

where $y_i = f(x_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$ represents Gaussian noise, we denote the input matrix by X, the output vector by $\mathbf{y} \in \mathbb{R}^n$, and the kernel matrix by $K(X, X) \in \mathbb{R}^{n \times n}$ with entries $K_{ij} = k(x_i, x_j)$.

For a test input x_*, we define

$$k(x_*, X) = [k(x_*, x_1), \ldots, k(x_*, x_n)].$$

The predictive distribution at x_* is Gaussian with mean and variance given by

$$\mu_n(x_*) = k(x_*, X)\left(K(X,X) + \sigma^2_{\text{noise}} I\right)^{-1} \mathbf{y}, \tag{3}$$

$$\sigma^2_n(x_*) = k(x_*, x_*) - k(x_*, X)\left(K(X,X) + \sigma^2_{\text{noise}} I\right)^{-1} k(X, x_*). \tag{4}$$

The posterior mean $\mu_n(x_*)$ is given by a weighted linear combination of the target values $\mathbf{y}$, where the weights are determined by the kernel values between the test input x_* and the training inputs X. In the posterior variance $\sigma^2_n(x_*)$, the first term $k(x_*, x_*)$ represents the prior variance at x_*, while the second term quantifies the reduction in uncertainty due to the observed data. The posterior variance decreases as x_* becomes closer to the training inputs.

The kernel function $k(x, x')$ defines the covariance between the function values at two input points x and x', encoded by their proximity in the input space (e.g., antibody sequence similarity). In our study, when treating antibody sequences as categorical inputs, we implemented GP surrogate with three different kernel functions (SSK, TOK, HED) using the default settings of the AntBO GP framework (Khan et al. (2023)) (`https://github.com/huawei-noah/HEBO/tree/master/AntBO`), implemented with `GPyTorch` (Gardner et al. (2021)) (v1.12).

C. Kernel Functions and Embedding

As antibody sequences are composed of categorical symbols forming strings, a GP surrogate model requires a kernel function specifically designed for string inputs. We describe two kernel functions, and the Hamming embedding via dictionaries used in our experiments.

C.1. Sub-sequence String Kernel (SSK)

The SSK quantifies the similarity between two strings by counting the number of shared subsequences of a specified length. This method is particularly well suited for unaligned antibody sequences, which are inherently represented as strings and can vary in length. The SSK effectively captures information where the sequential order of amino acids is consistent, even when exact matching is not necessary, as it accounts for gaps between amino acids. The nth-order SSK between two sequences a and b is defined as Moss et al. (2020):

$$k_n(a, b) = \sum_{u \in \Sigma^n} c_u(a)\, c_u(b), \tag{5}$$

$$c_u(s) = \lambda_m^{|u|} \sum_{1 < i_1 < \cdots < i_{|u|} < |s|} \lambda_g^{(i_{|u|} - i_1)} I_u\big((s_{i_1}, \ldots, s_{i_{|u|}})\big), \tag{6}$$

where Σ^n represents the set of all possible subsequences of length n drawn from the amino acid pool Σ. The indicator function $I_x(y)$ evaluates to 1 if the subsequence x appears in sequence y, and 0 otherwise. The match decay hyperparameter $\lambda_m \in [0, 1]$ and the gap decay hyperparameter $\lambda_g \in [0, 1]$ control the penalization of gaps and the influence of matched sequences.

For example, as illustrated in Figure 2, the SSK-based similarity between `EVQLEA` and `ELVLEE` is given by:

$$\begin{aligned} k(\texttt{EVQLEA}, \texttt{ELVLEE}) = &(\lambda_m^2)(\lambda_m^2 \lambda_g^1) \\ &+ (\lambda_m^3 \lambda_g^1)(\lambda_m^3 \lambda_g^1) \\ &+ (\lambda_m^2 \lambda_g^3)\left(\lambda_m^2 (1 + \lambda_g^3 + \lambda_g^4)\right). \end{aligned} \tag{7}$$

For fair comparison across sequences of varying lengths, the kernel is normalized as follows:

$$\tilde{K}_n(a,b) = \frac{k_n(a,b)}{\sqrt{k_n(a,a)\,k_n(b,b)}}. \tag{8}$$

In practice, a vectorized implementation (Beck and Cohn (2017)) leveraging parallel computation significantly improves the computational efficiency of optimizing kernel hyperparameters, enabling fine-grained tuning using gradient-based methods. In our study, the SSK was implemented using the `GAUCHE` library (Griffiths et al. (2023)) (`https://github.com/leojklarner/gauche`), enabling its execution within the GPU-compatible `GPyTorch` (Gardner et al. (2021)) (v1.12) environment.

C.2. Transformed Overlap Kernel (TOK)

The overlap (or Hamming) kernel (Ru et al. (2020)) is a kernel function that measures sequence similarity by counting the number of positions at which two sequences share the same amino acid. Given two strings $a = (a_1, \ldots, a_d)$ and $b = (b_1, \ldots, b_d)$, the overlap kernel is defined as

$$K(a,b) = \frac{\sigma}{d} \sum_{i=1}^{d} \delta(a_i, b_i), \tag{9}$$

where d denotes the maximum sequence length, σ represents the lengthscale parameter, and $\delta(a_i, b_i)$ is the Kronecker delta function, which equals 1 if $a_i = b_i$ and 0 otherwise.

The transformed overlap kernel is a modified version of the overlap kernel, offering greater expressiveness by learning different lengthscales for each input dimension. It is defined as

$$k_{\text{TOK}}(a,b) = \exp\left(\frac{1}{d} \sum_{i=1}^{d} l_i\, \delta(a_i, b_i)\right), \tag{10}$$

where l_i is the lengthscale at the ith position. Using the AntBO framework (Khan et al. (2023)), the sequence similarity is computed using the transformed overlap kernel and subsequently processed through a radial basis function (RBF) kernel.

C.3. Hamming Embedding via Dictionaries (HED)

HED is a technique designed to efficiently explore discrete or categorical search spaces through a randomized procedure (Deshwal et al. (2023)). A random dictionary matrix $A \in \mathbb{R}^{m \times d}$ is constructed diversely using different bias parameters (θ) from a Bernoulli distribution over 20 categories, representing the 20 standard amino acids. The dimension d corresponds to the maximum sequence length within the sequence pool, yielding categorical inputs in the space $\{0, 1, \ldots, 19\}^d$. The number of basis vectors m is set to 64, which defines the cardinality of the embedded input space. Antibody sequences are converted to an ordinal representation z, and subsequently projected into a latent space via an affine transformation using the random dictionary matrix A.

$$2\phi_A(z) = d\mathbf{1}_m - \bar{A}\bar{z}, \tag{11}$$

where $\bar{a}_{ij} = 2a_{ij} - 1$ and $\bar{z}_i = z_i - 1 \in \{-1, 1\}$. Using this formulation, the embedding distance between input pairs is derived as

$$2\|\phi_A(z) - \phi_A(z')\|_2 = \|\bar{A}\bar{r}\|_2, \tag{12}$$

where $\bar{r} = \bar{z} - \bar{z}'$. A Matérn-5/2 kernel with automatic relevance determination (ARD) is then applied to control the trade-off between model complexity and smoothness.

D. Fixed Feature LLM Surrogate (Bayesian Linear Probing)

Ab-RoBERTa was used as a frozen feature extractor, with its transformer weights remaining unchanged during training. The regression head is trained via Bayesian linear regression, referred to as Bayesian linear probing. The same hyperparameters as in the adaptive feature LLM surrogate were used, except that the LoRA module was omitted

Algorithm 2 Ab-L3BO – using Ab-RoBERTa as a feature extractor

Require: Objective function f, pre-trained LLM feature extractor ϕ_W, surrogate model $\hat{f}_\theta$, feasible set of antibody sequences $D_{\text{cand}} = \{x_i\}_{i=1}^n$, initial dataset $D_1 = \{(x_i, f(x_i))\}_{i=1}^m$, number of evaluations T, number of nearest neighbors κ

for $t = 1, \ldots, T$ **do**
  Construct feature dataset: $\phi_t = \{(\phi_W(x), f(x)) : x, f(x) \in D_t\}$
  Compute posterior approximation: $p(g_t \mid D_t) = LA(g_\theta, \phi_t)$
  Construct $KNN_t \subseteq D_{\text{cand}}$ containing κ nearest neighbors around the current best point $x^* = \arg\min_{x \in D_t} \hat{f}_t(x)$
  Optimize acquisition function over KNN_t: $x_t = \arg\min_{x \in KNN_t} \alpha(x \mid D_t)$
  Evaluate the reward $f(x_t)$
  Augment the data: $D_{t+1} = D_t \cup \{(x_t, f(x_t))\}$
  Update the feasible set: $D_{\text{cand}} = D_{\text{cand}} \setminus \{x_t\}$
end for
return $\arg\min_{(x, f(x)) \in D_{T+1}} f(x)$

E. LoRA and LA Configuration

Table 1: LoRA configuration used in the Ab-RoBERTa model

Parameter	Value
Rank	16
LoRA alpha	8
Target modules	query, value, key
LoRA dropout	16
Bias	None
Init LoRA weights	Gaussian
Module to save	head

Table 2: LA configuration used in the Ab-RoBERTa model

LA configuration	
Noise variance	1.00×10^{-3}
Hessian factorization	kron
Subset of weights	all
Marginal likelihood	posthoc
Prior prediction structure	layerwise
Batch size	20
Epochs	500
Weight decay	5.00×10^{-4}
Learning rate	1.00×10^{-3}

F. The Performance of Ab-L3BO Was Enhanced Through Model Optimization

To assess the performance of Ab-L3BO, we evaluated it using the AAYL49 antibody dataset (Engelhart et al. (2022)), which consists of 19,401 antibody sequences paired with their respective binding affinity. Based on data from 53 FDA-approved monoclonal antibodies (Table 3), more than 80% of antibodies exhibit an equilibrium dissociation constant (K_D) of less than 1 $\log_{10}$ nM. From this observation, we categorized the antibodies in the AAYL49 dataset into three groups based on affinity strength: strong, medium, and weak (Figure 3).

A subset of 50 randomly selected antibodies with K_D values ranging from 2 to 3 $\log_{10}$ nM (weak-affinity group) was used to investigate the impact of (1) the LLM surrogate model, and (2) the candidate sampling strategy.

Although the Ab-RoBERTa effectively captured general antibody features from sequences, it did not incorporate antigen-specific binding affinity information during training. To assess whether the Ab-RoBERTa can provide useful information relevant to antigen-specific binding affinity, we compared two LLM surrogates as follows. The fixed-feature LLM surrogate (Bayesian linear probing) (Appendix D and Algorithm 2) maintains frozen transformer weights, whereas the adaptive-feature LLM surrogate (Bayesian LoRA) (Algorithm 1) updates attention layer weights in each iteration of the Ab-L3BO. Our results show that after 50 steps, the lowest K_D achieved by the adaptive-feature LLM surrogate was 8-fold lower (t-test p-value = 0.0008) than that of the fixed-feature LLM surrogate. This finding indicates that the integration of binding affinity information into Ab-RoBERTa enhances the optimization of antigen-specific antibody sequences (Figure 3).

Additionally, previous studies have shown that BO performance can be enhanced through local probabilistic approach for global optimization (Khan et al. (2023); Wan et al. (2021); Kristiadi et al. (2024); Eriksson et al. (2020)). Building on this concept, we performed a local search for sequence candidates within their embedding space in the boundary of k-nearest neighbors ($k = 5000$) centered around the current best sequence. As a result, the lowest K_D achieved through local search was 3.6-fold lower (t-test $p = 0.0048$) than that obtained from random candidate selection, clearly demonstrating the effectiveness of this approach over random selection. (Figure 3).

Table 3: FDA-approved monoclonal antibodies and their binding affinity (K_D).

mAb	Brand name	Indication	Year	K_D (nM)	$\log_{10} K_D$ (nM)
Abciximab	Reopro	Prevention of blood clots in angioplasty	1994	5	0.699
Adalimumab	Humira	Rheumatoid arthritis	2002	0.0766	-1.116
Atezolizumab	Tecentriq	Bladder cancer	2016	0.43	-0.367
Avelumab	Bavencio	Merkel cell carcinoma	2017	0.042	-1.376
Benralizumab	Fasenra	Asthma	2017	0.011	-1.959
Bevacizumab	Avastin	Colorectal cancer	2004	10.9	1.037
Bezlotoxumab	Zinplava	Prevention of Clostridium difficile infection recurrence	2016	1	0.000
Brentuximab	Adcetris	Hodgkin lymphoma systemic anaplastic large cell lymphoma	2011	2	0.301
Brodalumab	Siliq Lumicef	Plaque psoriasis	2017	0.239	-0.622
Canakinumab	Ilaris	Muckle-Wells syndrome	2009	1.07	0.029
Cemiplimab	Libtayo	Cutaneous squamous cell carcinoma	2018	0.6	-0.222
Certolizumab	Cimzia	Crohn's disease	2008	0.09	-1.046
Cetuximab	Erbitux	Colorectal cancer	2004	0.39	-0.409

mAb	Brand name	Indication	Year	K_D (nM)	$\log_{10} K_D$ (nM)
Crizanlizumab	Adakveo	Sickle cell disease	2019	12.4	1.093
Daratumumab	Darzalex	Multiple myeloma	2015	3.3	0.519
Denosumab	Xgeva Prolia	Bone loss	2010	0.003	-2.523
Dupilumab	Dupixent	Atopic dermatitis	2017	0.00476	-2.322
Durvalumab	Imfinzi	Bladder cancer	2017	0.2	-0.699
Elotuzumab	Empliciti	Multiple myeloma	2015	40.4	1.606
Erenumab	Aimovig	Migraine prevention	2018	0.02	-1.699
Evolocumab	Repatha	High cholesterol	2015	0.004	-2.398
Galcanezumab	Emgality	Migraine prevention	2018	0.031	-1.509
Golimumab	Simponi	Rheumatoid and psoriatic arthritis ankylosing spondylitis	2009	0.018	-1.745
Guselkumab	Tremfya	Plaque psoriasis	2017	0.035	-1.456
Ibalizumab	Trogarzo	HIV infection	2018	0.0825	-1.084
Infliximab	Remicade	Crohn disease	1998	13.6	1.134
Ipilimumab	Yervoy	Metastatic melanoma	2011	5.25	0.720
Ixekizumab	Taltz	Psoriasis	2016	0.0018	-2.745
Mepolizumab	Nucala	Severe eosinophilic asthma	2015	0.1	-1.000
Moxetumomab	Lumoxiti	Hairy cell leukemia	2018	1.38	0.140
Necitumumab	Portrazza	Non-small cell lung cancer	2015	10	1.000
Nivolumab	Opdivo	Melanoma non-small cell lung cancer	2014	1.45	0.161
Obiltoxaximab	Anthim	Prevention of inhalational anthrax	2016	0.33	-0.481
Obinutuzumab	Gazyva Gazyvaro	Chronic lymphocytic leukemia	2013	10	1.000
Ofatumumab	Arzerra	Chronic lymphocytic leukemia	2009	0.167	-0.777
Omalizumab	Xolair	Asthma	2003	7	0.845
Palivizumab	Synagis	Prevention of respiratory syncytial virus infection	1998	28	1.447
Panitumumab	Vectibix	Colorectal cancer	2006	0.05	-1.301
Pembrolizumab	Keytruda	Melanoma	2014	0.029	-1.538
Pertuzumab	Perjeta	Breast cancer	2012	0.87	-0.060
Ramucirumab	Cyramza	Gastric cancer	2014	0.05	-1.301
Ravulizumab	Ultomiris	Paroxysmal nocturnal hemoglobinuria	2018	0.5	-0.301
Raxibacumab	Abthrax	Anthrax infection	2012	2.78	0.444
Reslizumab	Cinqaero Cinqair	Asthma	2016	0.081	-1.092
Risankizumab	Skyrizi	Plaque psoriasis	2019	0.021	-1.678
Rituximab	MabThera Rituxan	Non-Hodgkin lymphoma	1997	4.69	0.671
Romosozumab	Evenity	Osteoporosis in postmenopausal women at increased risk of fracture	2019	0.011	-1.959
Sarilumab	Kevzara	Rheumatoid arthritis	2017	0.0128	-1.893
Secukinumab	Cosentyx	Psoriasis	2015	0.15	-0.824
Tildrakizumab	Ilumya	Plaque psoriasis	2018	0.297	-0.527
Tocilizumab	RoActemra Actemra	Rheumatoid arthritis	2010	2.5	0.398

mAb	Brand name	Indication	Year	K_D (nM)	$\log_{10} K_D$ (nM)
Trastuzumab	Herceptin	Breast cancer	1998	0.3	-0.523
Ustekinumab	Stelara	Psoriasis	2009	0.106	-0.975

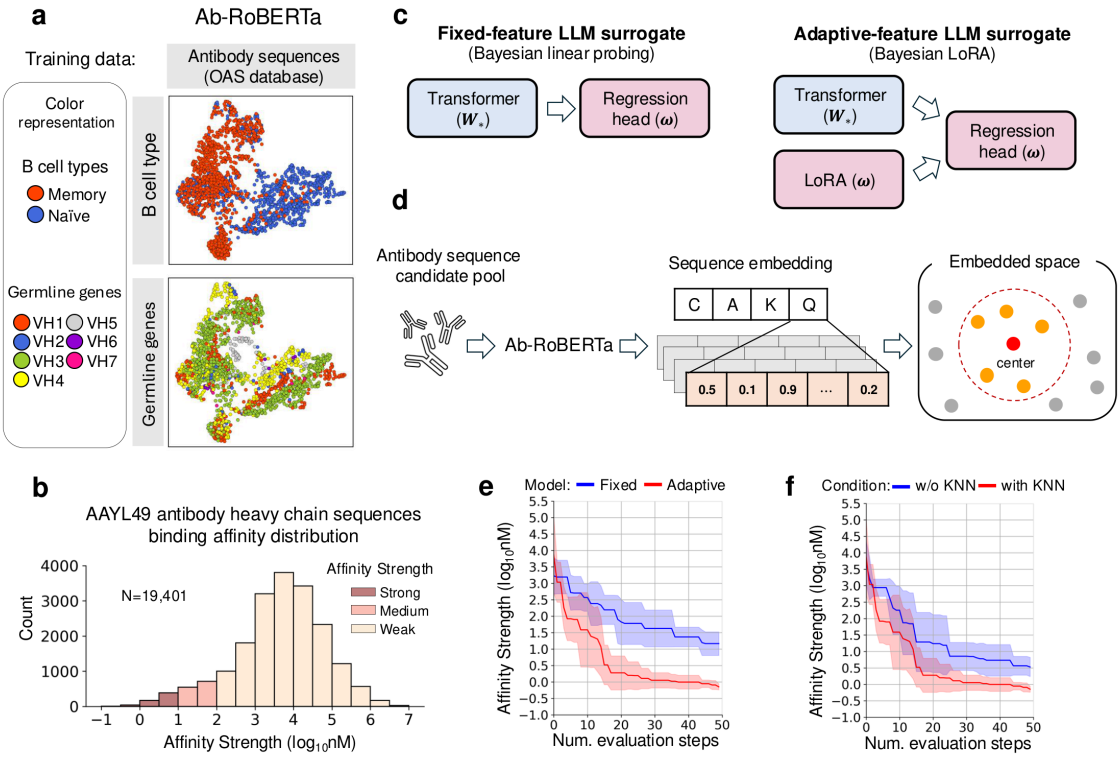
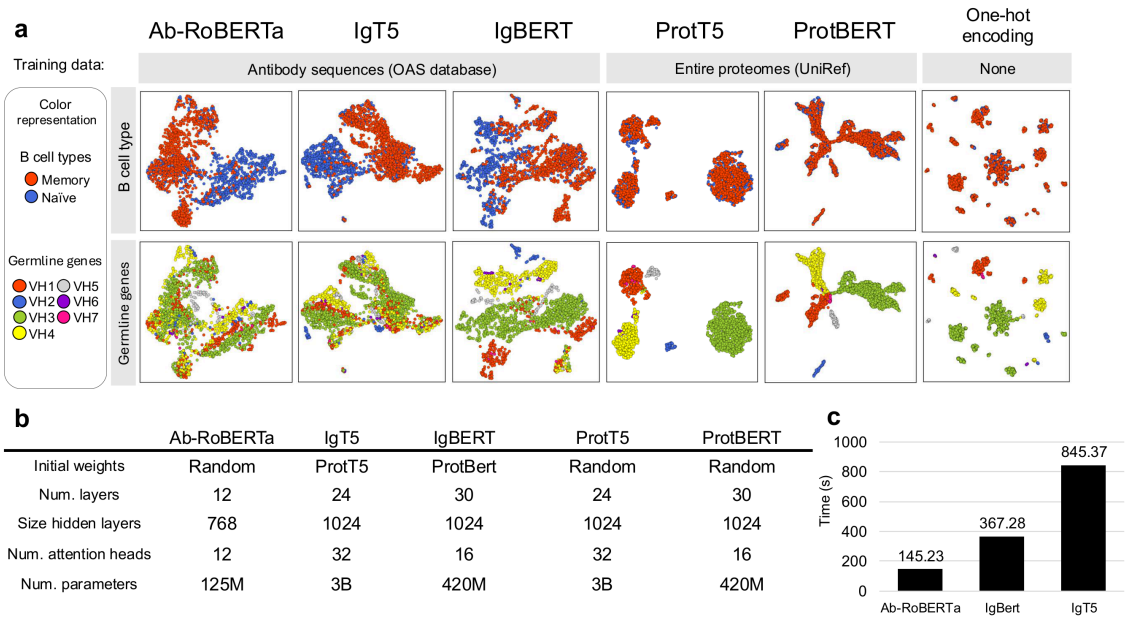


Figure 3: **The performance comparison using different sequence representation methods.** **(a)**, Ab-RoBERTa converts 2000 antibody heavy chain sequences into numerical embedding vectors and those embeddings are visualized using Uniform Manifold Approximation and Projection (UMAP). A single data point represents a single antibody sequence, and the color shows the properties of antibody sequences. **(b)**, The antibody binding affinity distribution is plotted from data using AAYL49 antibody database. **(c)**, Two LLM surrogate models applied in this study. The figure is regenerated from Kristiadi et al. (2024). In the fixed-feature LLM surrogate, the transformer weights remain fixed (blue), while only the weights in regression head are updated (pink). In contrast, in the adaptive-feature LLM surrogate, the original transformer weights remain fixed (blue), but the trainable LoRA and the weights in regression head are updated (pink). **(d)**, Antibody sequences in the candidate pool are transformed into contextual embeddings using Ab-RoBERTa. In the embedding space, the k-number of candidates are selected by k-nearest neighbor (KNN) centered around the current best sequence. **(e)** and **(f)**, The experiment was conducted over 50 steps (x-axis), and the lowest KD at each step plotted on the y-axis. This entire procedure was repeated in five times to compute mean (solid line) and standard deviation (shade area). **(e)**, The blue line represents the fixed-feature LLM with KNN-based sampling, while the red line represents the adaptive-feature LLM with KNN-based sampling. **(f)**, The graph compared random sampling and KNN-based candidate selection, applied within the adaptive-feature LLM surrogate model.

G. LLM Representation Comparison and Runtime Analysis for Antibody Sequence Optimization

We evaluated representations from five pre-trained LLMs by embedding 2,000 antibody heavy-chain sequences and visualizing the resulting representations using UMAP, with one-hot encoding as a baseline. Antibody-specific LLMs produced embeddings that more clearly separated biologically relevant antibody properties, indicating improved representation of immunological sequence features. We further report model configurations and benchmark the runtime of a single Bayesian optimization step under fixed experimental settings (batch size 30, 50 initial sequences, and 5,000 candidate samples).



	Ab-RoBERTa	IgT5	IgBERT	ProtT5	ProtBERT
Initial weights	Random	ProtT5	ProtBert	Random	Random
Num. layers	12	24	30	24	30
Size hidden layers	768	1024	1024	1024	1024
Num. attention heads	12	32	16	32	16
Num. parameters	125M	3B	420M	3B	420M

Figure 4: **Comparison between different pre-trained LLMs. (a)**, Five different pre-trained LLMs converts 2000 antibody heavy chain sequences into numerical embedding vectors. For comparison, one-hot encoding was used as the baseline representation. Those embeddings are projected using uniform manifold approximation and projection (UMAP). A single data point represents a single antibody heavy chain sequence, and the color shows the properties of antibody sequences. **(b)**, Configurations for the five different pre-trained LLMs are shown. **(c)**, The runtime for a single step of BO was measured under the conditions of a batch size of 30, 50 initial sequences, and 5,000 sampled candidate points.

H. Effect of Initial Sequence Set Size on Ab-L3BO Optimization

While larger datasets generally yield better performance, it is essential to establish practical guidelines for the initial sequence dataset size. To address this, we evaluated Ab-L3BO by gradually increasing the number of initial sequences. The optimization performance generally improved as the initial set size increased, and the performance with 500 initial sequences exhibited a statistically significant improvement. Although 500 initial sequences appear desirable for ideal optimization, this number may be impractical or inefficient in wet-lab experimental settings (Figure 5).

The optimal dataset size may vary depending on the specific dataset; however, starting with 50 initial sequences could be reasonable, as the mean of the lowest K_D across five iterations remained in the picomolar range with small variation (Figure 5). In addition, we repeated the analysis using

data sampled within the range of 3 to 5 $\log_{10}$ nM; however, there was no statistically significant impact on the composition of the initial input dataset .

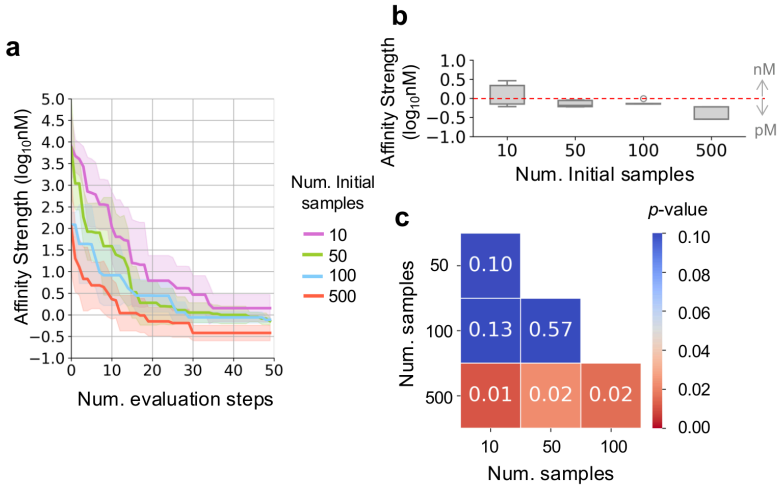


Figure 5: **The impact of the initial input dataset.** **(a)** The experiment was conducted with four different cases of initial sample sizes over 50 steps, and the lowest K_D at each step was plotted on the y-axis. The entire procedure was repeated five times to obtain the mean (solid line) and standard deviation (shaded area). **(b)** The final outputs from five iterations were plotted as a bar plot. The red line represents a K_D value of 1 nM. Values below this line fall within the picomolar range. **(c)** Pairwise t-test p-values corresponding to panel **(b)** are shown.

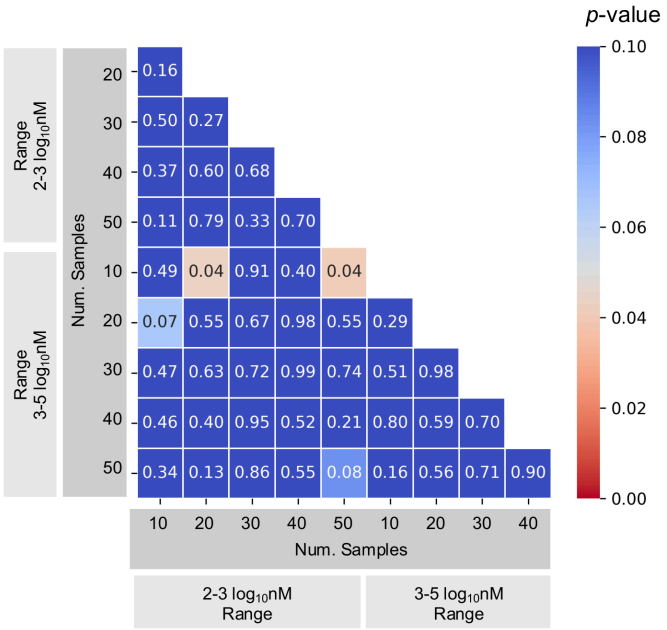


Figure 6: **A comparative analysis was performed across various sampling strategies.** The initial dataset comprised 10–50 data points drawn from two distinct groups within the weak binding range: 2–3 $\log_{10}$ nM and 3–5 $\log_{10}$ nM. Following five iterations of Ab-L3BO, the five lowest K_D values from each input set were identified and statistically compared using a t-test.

I. Ab-L3BO Generalizes Well Across Diverse Datasets

To evaluate the consistency of Ab-L3BO performance on an independent dataset, we applied Ab-L3BO to a dataset obtained from the Absolut! database (Robert et al. (2022)). This dataset, referred to as *1ADQ_CDRH3* for convenience, comprises 999,985 CDRH3 antibody sequences of varying lengths known to bind the IgG4 Fc region (PDB: 1ADQ). The binding affinity of each sequence was assessed using the Absolut! simulator (Robert et al. (2022)), which relies on the Miyazawa–Jernigan potential (Miyazawa and Jernigan (1996)) for its scoring mechanism. The simulator applied a sliding-window approach, evaluating affinity strength across every consecutive 11-residue segment within each CDRH3 sequence. This process identified the most critical 11-residue segment for each antibody. A refined set of non-redundant 11-residue segments was obtained by eliminating duplicate sequences, referred to as the *1ADQ_11mer* dataset. The binding affinity values for both datasets ranged from -109.3 to -56.9.

On the *1ADQ_CDRH3* dataset, Ab-L3BO demonstrated superior sampling efficiency compared to four other models, as measured by the mean across replicates, although the differences did not reach strong statistical significance. Similarly, when applied to the *1ADQ_11mer* dataset, the mean output of Ab-L3BO across five replicates achieved the most optimal values among the compared models. However, the statistical significance was comparable to that of the other models (Figure 7). Taken together with the results shown in Figure 2, these findings indicate that Ab-L3BO generalizes effectively across different datasets, while also exhibiting enhanced optimization capability, particularly when larger initial datasets are available as shown in Appendix H.

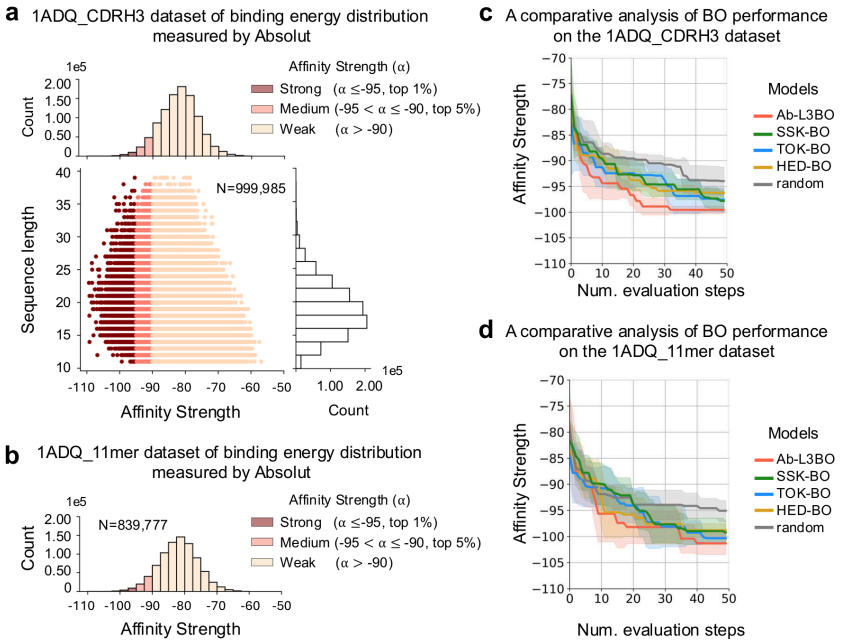


Figure 7: **Performance comparison across different BO models in application of Absolut! database.** **(a)**, The 1ADQ_CDRH3 dataset was characterized in terms of affinity strength and sequence length. The affinity strength from Absolut! simulator, ranged from -109.3 to -56.9, while the sequence length varied from 11 to 39. **(b)**, The 1ADQ_11mer dataset was similarly characterized, with affinity scores remaining within the same range -109.3 to -56.9. **(c)**, Five BO models were applied to the 1ADQ_CDRH3 dataset. For each model, the strongest affinity score was tracked and plotted over 50 steps. The solid line represents the mean across five iterations, and the shaded region indicates the corresponding standard deviation. **(d)**, In a parallel experiment, the same procedure was repeated on the 1ADQ_11mer dataset.