Leveraging in Silico Predictors for Antibody Design via Multi-Objective Bayesian Optimization

Jackie Rao[*]
MRC Biostatistics Unit, University of Cambridge, UK

Ferran Gonzalez Hernandez **Leon Gerard** **Alexandra Gessner**
Centre for Artificial Intelligence, Data Science and Artificial Intelligence, R&D
AstraZeneca, Barcelona, Spain

Abstract

Antibody lead optimization is inherently a multi-objective challenge in drug discovery. Achieving a balance between different drug-like properties is crucial for the development of viable candidates, and this search becomes exponentially challenging as desired properties grow. The ever-growing zoo of sophisticated *in silico* tools for predicting antibody properties calls for an efficient joint optimization procedure to overcome resource-intensive sequential filtering pipelines. We present BOAT, a versatile Bayesian optimization framework for multi-property antibody engineering. Our 'plug-and-play' framework couples uncertainty-aware surrogate modeling with a genetic algorithm to jointly optimize various predicted antibody traits while enabling efficient exploration of sequence space. Through systematic benchmarking against genetic algorithms, we demonstrate that our method efficiently explores the Pareto front where the combinatorial ground truth is available.

1 Introduction

Lead optimization is crucial for therapeutic antibody development, aiming to advance candidates into effective drugs by improving multiple properties such as affinity, cross-reactivity, manufacturability, stability, and immunogenicity. Modern *in silico* methods such as machine learning predictors and physics-based simulations enable rapid, cost-effective estimation of antibody properties ahead of experiments. However, integrating predictions for several objectives is challenging: predictors are computationally expensive, property conflicts can limit sequence diversity, and the search space is vast. Thus, approaches that jointly optimize multiple objectives while quantifying uncertainty are critical for guiding experimental design and prioritizing promising antibody leads.

We address these challenges with BOAT (Bayesian optimization for antibody traits), a versatile multi-objective Bayesian optimization framework for antibody sequences. BOAT supports easy interfacing of arbitrary *in silico* predictors. This allows users to jointly leverage their preferred scoring functions for evaluating qualities of the candidate antibody and thus respond to particular requirements posed in a lead optimization campaign. We perform rigorous benchmarking of BOAT against genetic baselines and demonstrate that our method efficiently explores the Pareto front and fosters sequence diversity. While we focus here on antibodies, the approach extends naturally to proteins and peptides. In this work we consider models for binding affinity, humanness and naturalness.

[*] This work has been carried out during an internship at AstraZeneca, Cambridge, UK

2 Related Work

Traditional and Evolutionary Baselines Evolutionary algorithms have long been applied to discrete protein sequence optimization (Katoch et al., 2020), extending to multi-objective cases with algorithms like NSGA-II (Deb et al., 2002). Their inefficiency in high-dimensional protein spaces (Turner et al., 2021) led to specialized variants such as AdaLead (Sinai et al., 2020) and PEX (Ren et al., 2022), which leverage domain knowledge and wet-lab feedback. Integration with machine learning allows selection of beneficial mutations and diversity enhancement (Nigam et al., 2020; Yang et al., 2019; Nana Teukam et al., 2024); these *in silico* methods mirror directed evolution strategies. Surrogate-assisted approaches employ learned models to evaluate candidate sequences heuristically.

Generative Models and Reinforcement Learning (RL) Recent generative models, especially transformer-based and autoregressive PLMs (Rives et al., 2021; Ferruz et al., 2022), enable antibody sequence synthesis and conditional infilling of regions like CDRs (Shuai et al., 2023; Melnyk et al., 2023). Diffusion models are also emerging in protein design (He et al., 2024). These models can be guided towards desired properties using *in silico* predictors (Goel et al., 2024; Yang et al., 2025). RL offers sequential optimization using policies (models) rewarded by property predictors—see (Angermueller et al., 2020; Lee et al., 2025a) for PPO applications, and recent adaptations of Direct Preference Optimization (DPO) for single-objective protein design (Widatalla et al., 2024; Zhou et al., 2024). Multi-objective RL extensions tune generators with additional reward constraints (Ren et al., 2024).

Bayesian Optimization (BO) BO uses uncertainty-aware surrogates for sample-efficient improvement, commonly with Gaussian processes. Most frameworks target single objectives (González-Duque et al., 2024), struggling with high-dimensional, discrete sequence spaces (Wang et al., 2013). Latent space BO relies on VAE- or DAE-learned embeddings (Gómez-Bombarelli et al., 2018; Stanton et al., 2022), though verifying decoded sequences is challenging (Lee et al., 2025b). Sequence-space BO employs specialized kernels (e.g., BOSS (Moss et al., 2020), AntBO (Khan et al., 2022)) and trust region techniques (Eriksson et al., 2019). Hybrid generative-BO approaches, like CloneBO (Amin et al., 2024), combine language models and Bayesian sampling.

Multi-Objective Optimization Few methods optimize multiple objectives directly, usually quantifying Pareto front quality via hypervolume. RL variants add objectives as constraints (Ren et al., 2024), and gradient-based sequence optimizers require differentiable predictors (Luo et al., 2025; Emami et al., 2023). LaMBO extends BO with DAEs and generative infilling for multi-objective search (Stanton et al., 2022; Gruver et al., 2023); ALLM-Ab uses active-learning for hypervolume maximization (Furui and Ohue, 2025). Our work is the first to apply multi-objective BO for embedded discrete sequence space, optimizing black-box *in silico* predictors to search for Pareto-optimal antibodies.

3 Materials and Methods

3.1 Multi-Objective Bayesian Optimization

Bayesian Optimization (BO) is a sample-efficient active learning framework for global optimization of function $f : \mathcal{X} \to \mathbb{R}$, where f is often expensive to evaluate and lacks structure (eg. closed-form gradients) that would make it amenable to direct optimization methods (Frazier, 2018; Garnett, 2023). Given a dataset of previous (potentially noisy) evaluations $\mathcal{D}_t = (\mathbf{x}_i, y_i)_{i=1,\dots,t}$, a probabilistic surrogate model $p(f|\mathcal{D}_t)$ – usually a Gaussian process (Rasmussen and Williams, 2005) – is fit to this dataset which captures the current belief about the unknown objective function f. The sequential evaluation policy is encoded through an acquisition function that quantifies the utility of evaluating f at a candidate input point $\mathbf{x}$ and balances exploitation and exploitation of the objective. For single-objective optimization, a common choice for the acquisition function is expected improvement. Multi-objective optimization requires adapting the acquisition function in order to capture that trade-off between objectives. We use Expected Hypervolume Improvement (EHVI) (Emmerich et al., 2011) and its noisy extension, Noisy Expected Hypervolume Improvement (NEHVI) (Daulton et al., 2021). These objective functions promote expansion of the Pareto front and maximization of the associated hypervolume. We note that other multi-objective alternatives exist such as MORBO

(Daulton et al., 2022) and ParEGO (Knowles, 2006). Our implementation leverages the BOTorch framework (Balandat et al., 2020) - which also allows batch extensions of the acquisition functions above (Daulton et al., 2020) - whose modular design enables straightforward extension to additional acquisition functions.

3.2 BO in Sequence Space

Common kernels for Gaussian processes map from $\mathbb{R}^d \times \mathbb{R}^d$ or a subset thereof to the real line. In order to apply Bayesian optimization to sequences of amino acids defined by strings $s \in \mathcal{S}$, there are two options, 1) to define a string kernel that operates on string space directly, or 2) to embed the sequences to represent them in a numerical space. We choose the latter approach and consider one-hot and BLOSUM sequence encodings, detailed in Appendix A.1.

The embedding space is quite large for both considered embeddings; both give rise to sequence embeddings of size sequence length × number of amino acids (i.e., 20). We therefore employ a Gaussian process model that has been designed for this kind of high-dimensional problem, using the Tanimoto kernel (Ralaivola et al., 2005).

3.3 Genetic Optimizer

While for some embeddings, such as one-hot and BLOSUM, sequences can be reconstructed from an embedding, they still represent discrete objects which prevent us from using a gradient-based optimizer to optimize the acquisition function in embedding space. To overcome this limitation, we resort to a genetic algorithm (GA) for generating sequences guided by the acquisition score. In each iteration of the BO loop, we generate an initial population by slightly mutating the previously evaluated sequences – this way we ensure not to start in local minima of the acquisition function. We repeatedly mutate a population of sequences through mutation, crossover and recombination: details can be found in Appendix A.2.

3.4 Oracles

Affinity predictor We train a neural network predictor on experimental affinity data for each considered antibody-antigen pair to predict the delta in binding affinity with respect to the parental. To deal with the small number of data points, we augment the dataset by considering the difference in affinity between sequence pairs, inspired by Lin et al. (2025). Our model uses an AbLang-2 tokenizer Olsen et al. (2024) and a CNN-based regression head.

Humanness score We use `promb`'s implementation of the OASis score (Prihoda et al., 2022), a humanness score based on 9-mer peptide search in the Observed Antibody Space (OAS) (Kovaltsuk et al., 2018).

Sequence likelihoods We compute the mean log-probability of amino acids in sequences using the protein language model ESM-2 with 3B parameters (Lin et al., 2023).

Antibody lead optimization campaigns may target different sets of properties; one campaign might focus more on developability, while another might target cross-reactivity to multiple antigens, requiring a complete disparate set of *in silico* predictors. BOAT provides a simple scoring function interface that makes interchanging scoring functions straightforward.

4 Experiment: Cross-reactivity of a V_{HH}

We ran BOAT on a therapeutic nanobody to demonstrate the practical applicability of our framework to real-world antibody design scenarios. The lead optimization objective is to introduce cross-reactivity to two similar antigens, i.e., to enhance binding affinity on both while retaining or improving developability properties. We systematically optimize CDR1, CDR2, and CDR3 regions of the heavy chain individually, allowing up to 5 mutations per CDR region. The mutation space for each position was constrained to a curated dictionary of amino acids based on those in the experimental training data and structural considerations (e.g. fixing cysteines). We progressively increase the number of objectives from 2 to 4 to evaluate the scalability of BOAT with problem dimensionality and compare sequential and batch design. For the main task of introducing cross-reactivity, we leverage two affinity predictors described in Section 3.4 that were trained on the experimentally measured affinities for both antigens of 340 single-point and 26 quadruple mutations. We add the humanness score and a PLM likelihood as additional objectives (cf. Section 3.4).

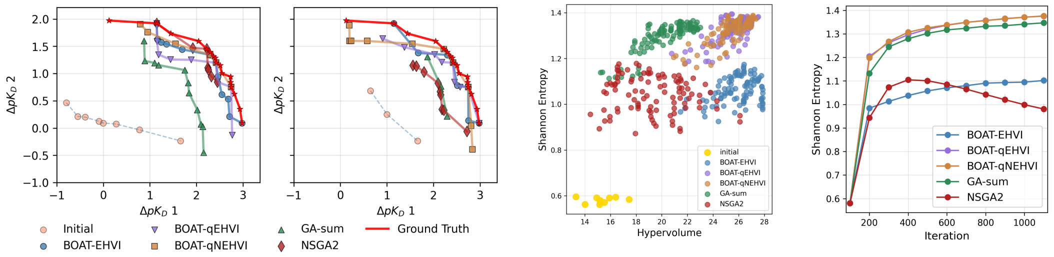
We benchmarked against two GA baselines. Our first GA baseline was set up as a standard GA (described in Section 3.3) which optimizes a normalized sum of the objectives. Our second GA baseline is NSGA-II (Deb et al., 2002). Due to the constrained mutation dictionary and the fast objective functions, we pre-computed a 'ground truth' Pareto front by scoring all possible CDR mutations for each CDR. We therefore evaluate our methods by comparing their discovered Pareto fronts to the ground truth, and track how the hypervolume evolves over oracle calls. The total number of 'ground truth' sequences are 1,438,121, 33,829,027 and 61,602,147 for CDR1, CDR2 and CDR3 respectively. All methods have a budget of 1000 oracle calls. Further experimental details can be found in Appendix B.

4.1 2 objectives

Initially we consider only the two affinity predictors as optimization objectives. In addition to the acquisition functions for sequential design (EHVI) and batch design (qEHVI), we include the batch version for noisy objectives (qNEHVI) in this setup. Since qNEHVI scales super-polynomially with the number of objectives (Daulton et al., 2021) and did not outperform qEHVI significantly in our experiments, we excluded it from further evaluation to maintain computational tractability.

In 2 dimensions, we can directly visualize the discovered Pareto front against the 'ground truth' Pareto front derived from exhaustive evaluation. A subset of these plots is visualized in Figure 1a, displaying the seed with the highest hypervolume among all GA methods and the seed with the highest hypervolume among all BOAT variants for that CDR. Plots for all seeds and also plots for earlier points in all the models can be found in the Appendix. BOAT successfully explores regions close to the true Pareto front, and can frequently find some of the true Pareto optimal sequences - even when searching a space of more than 63 million sequences.

Figure 3 shows the hypervolume evolution as a function of oracle calls, revealing that BOAT consistently achieves larger final hypervolumes compared to baseline methods while converging more rapidly. There was not a significant difference in the final hypervolume found between different acquisition functions for BOAT, but in general, batch acquisition functions tend to prioritize exploration early-on and therefore have a lower hypervolume in earlier iterations.



(a) Pareto fronts comparison (b) Diversity vs. HV (c) Evolution of diversity

Figure 1: CDR3 optimization results for 2-objective design with 5 mutations. (a) Discovered vs. ground truth Pareto fronts showing best-performing seed for GA methods and BOAT. Ground truth fronts obtained via exhaustive evaluation. (b) Scatter plot of hypervolume versus Shannon entropy for all seeds, methods, and every 100 iterations. Initial solutions highlighted in gold. (c) Evolution of Shannon entropy over optimization iterations, averaged over 10 seeds with standard error bands.

To assess the diversity of solutions discovered by each method, we computed the average Shannon entropy for all generated sequences for every 100 generated sequences, for all methods. A visualization of the results for CDR3 can be seen in Figure 1b, comparing Shannon entropy to hypervolume and iteration, with additional figures for other CDRs in the Appendix. We see that notably, the batch acquisition methods (BO-qEHVI and BO-qNEHVI) achieve the optimal combination of both high hypervolume performance and high sequence diversity; this superiority over all other methods is even more pronounced for CDR1 and CDR2. The larger diversity likely stems from the batch acquisition's inherent mechanism of selecting multiple diverse candidates simultaneously, naturally promoting exploration of different regions of the sequence space. BO-EHVI, while achieving competitive hypervolume performance, exhibits lower sequence diversity, suggesting more focused exploitation around promising regions. While both GA methods are inferior in hypervolume performance, it is interesting that GA-sum is able to explore more diverse sequences compared to NSGA-II. High sequence diversity is crucial for experimental validation campaigns, as it provides multiple distinct candidates for testing while maintaining optimization quality. In Figure 1c, we see additionally that BOAT successfully maintains sequence diversity throughout the algorithm.

4.2 3 objectives

We then added a third objective to the two binding affinity predictors: the OASis humanness predictor. Figure 2 shows the hypervolume evolution compared to oracle calls, showing that BOAT outperformed both of the GA baselines in both the versions of EHVI. We look at 4 objectives in Appendix C.3.

To evaluate whether our multi-objective optimization approach leads to improved fitness beyond the explicitly optimized objectives, we scored the first 300 generated sequences from each method using ESM-2. Figure 7 in Appendix C.2 reveals that BOAT automatically generates sequences with slightly higher PLM scores early in the optimization process, even without explicitly optimizing for this objective. Multi-objective Bayesian optimization not only excels at the specified objectives but also can implicitly optimize for sequence properties that correlate with biological fitness.

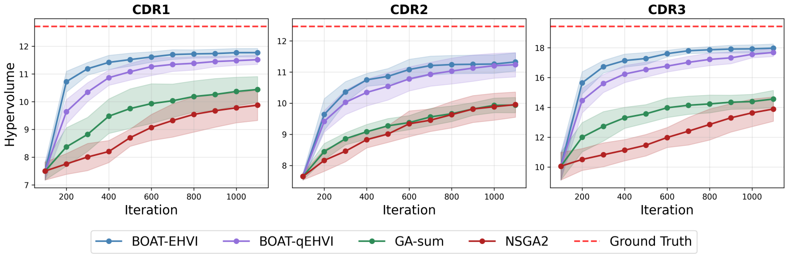


Figure 2: Hypervolume evolution for 3-objective CDR optimization. Results averaged over 10 seeds with standard error.

5 Discussion

Real-world antibody development demands simultaneous optimization of multiple properties with small experimental budgets and long timescales. In this work, we presented BOAT, a lightweight plug-and-play multi-objective Bayesian optimization framework for antibody lead optimization that enables efficient exploration of sequence space to optimize a Pareto front. BOAT allows users to leverage existing state-of-the-art *in silico* tools for antibody property prediction while efficiently exploring Pareto-optimal tradeoffs between up to four competing objectives. Future work will explore techniques from many-objective optimization to overcome the increased complexity of computing the Pareto front that quickly becomes problematic as more objectives are added (Ishibuchi et al., 2008). Additionally, we plan to investigate systematic diversity promotion mechanisms for sequence generation, as we encountered practical limitations imposed by the genetic algorithm that tends to converge prematurely and become over-saturated with sequences from a small subset of the sequence space. We further hope to enhance model expressivity and performance in the future with more tailored GP models.

Antibody design remains a laborious interplay between wetlab experiments and computationally driven design, an effort that BOAT effectively streamlines, making it a viable solution for lab-in-the-loop antibody design.

References

Amin, A. N., Gruver, N., Kuang, Y., Li, L., Elliott, H., McCarter, C., Raghu, A., Greenside, P., and Wilson, A. G. (2024). Bayesian optimization of antibodies informed by a generative model of evolving sequences.

Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K., and Colwell, L. (2020). Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). Botorch: A framework for efficient monte-carlo bayesian optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21524–21538. Curran Associates, Inc.

Blank, J. and Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509.

Daulton, S., Balandat, M., and Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864. Curran Associates, Inc.

Daulton, S., Balandat, M., and Bakshy, E. (2021). Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2187–2200. Curran Associates, Inc.

Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2022). Multi-objective Bayesian optimization over high-dimensional search spaces. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 507–517. PMLR. ISSN: 2640-3498.

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

Emami, P., Perreault, A., Law, J., Biagioni, D., and St. John, P. (2023). Plug & play directed evolution of proteins with gradient-based discrete MCMC. *Machine Learning: Science and Technology*, 4(2):025014.

Emmerich, M. T. M., Deutz, A. H., and Klinkenberg, J. W. (2011). Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, page 2147–2154. IEEE.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local Bayesian optimization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ferruz, N., Schmidt, S., and Höcker, B. (2022). ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1).

Frazier, P. I. (2018). A tutorial on Bayesian optimization.

Furui, K. and Ohue, M. (2025). ALLM-Ab: Active learning-driven antibody optimization using fine-tuned protein language models.

Garnett, R. (2023). Bayesian Optimization.

Gessner, A., Ober, S. W., Vickery, O., Oglić, D., and Uçar, T. (2024). Active learning for affinity prediction of antibodies.

Goel, S., Thoutam, V., Marroquin, E. M., Gokaslan, A., Firouzbakht, A., Vincoff, S., Kuleshov, V., Kratochvil, H. T., and Chatterjee, P. (2024). Memdlm: De novo membrane protein design with masked discrete diffusion protein language models.

González-Duque, M., Michael, R., Bartels, S., Zainchkovskyy, Y., Hauberg, S., and Boomsma, W. (2024). A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences. arXiv:2406.04739 [cs].

Gruver, N., Stanton, S., Frey, N. C., Rudner, T. G. J., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. (2023). Protein design with guided discrete diffusion. arXiv:2305.20009 [cs].

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276. Publisher: American Chemical Society.

He, X.-h., Li, J.-r., Xu, J., Shan, H., Shen, S.-y., Gao, S.-h., and Xu, H. E. (2024). AI-driven antibody design with generative diffusion models: current insights and future directions. *Acta Pharmacologica Sinica*, 46(3):565–574.

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

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, pages 2419–2426. IEEE.

Katoch, S., Chauhan, S. S., and Kumar, V. (2020). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126.

Khan, A., Cowen-Rivers, A. I., Grosnit, A., Deik, D.-G.-X., Robert, P. A., Greiff, V., Smorodina, E., Rawat, P., Dreczkowski, K., Akbar, R., Tutunov, R., Bou-Ammar, D., Wang, J., Storkey, A., and Bou-Ammar, H. (2022). Antbo: Towards real-world automated antibody design with combinatorial Bayesian optimisation.

Knowles, J. (2006). ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

Kovaltsuk, A., Leem, J., Kelm, S., Snowden, J., Deane, C. M., and Krawczyk, K. (2018). Observed antibody space: a resource for data mining next-generation sequencing of antibody repertoires. *The Journal of Immunology*, 201(8):2502–2509.

Lee, C. S., Hayes, C. F., Vashchenko, D., and Landajuela, M. (2025a). Reinforcement learning for antibody sequence infilling.

Lee, S., Park, J., Chu, J., Yoon, M., and Kim, H. J. (2025b). Latent Bayesian optimization via autoregressive normalizing flows. arXiv:2504.14889 [cs].

Lin, J. Y.-Y., Hofmann, J. L., Leaver-Fay, A., Liang, W.-C., Vasilaki, S., Lee, E., Pinheiro, P. O., Tagasovska, N., Kiefer, J. R., Wu, Y., Seeger, F., Bonneau, R., Gligorijevic, V., Watkins, A., Cho, K., and Frey, N. C. (2025). DyAb: sequence-based antibody design and property prediction in a low-data regime. *bioRxiv*.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130.

Luo, J., Ding, K., and Luo, Y. (2025). Pareto-optimal sampling for multi-objective protein sequence design. *iScience*, 28(3):112119.

Melnyk, I., Chenthamarakshan, V., Chen, P.-Y., Das, P., Dhurandhar, A., Padhi, I., and Das, D. (2023). Reprogramming pretrained language models for antibody sequence infilling. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24398–24419. PMLR.

Moss, H., Leslie, D., Beck, D., González, J., and Rayson, P. (2020). BOSS: Bayesian optimization over string spaces. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15476–15486. Curran Associates, Inc.

Nana Teukam, Y. G., Zipoli, F., Laino, T., Criscuolo, E., Grisoni, F., and Manica, M. (2024). Integrating genetic algorithms and language models for enhanced enzyme design. *Briefings in Bioinformatics*, 26(1).

Nigam, A., Friederich, P., Krenn, M., and Aspuru-Guzik, A. (2020). Augmenting genetic algorithms with deep neural networks for exploring the chemical space. In *Proceedings of the International Conference on Learning Representations*.

Oglic, D. and Gärtner, T. (2018). Learning in reproducing kernel Krein spaces. In *Proceedings of the 35th International Conference on Machine Learning*.

Oglic, D. and Gärtner, T. (2019). Scalable learning in reproducing kernel Krein spaces. In *Proceedings of the 36th International Conference on Machine Learning*.

Olsen, T. H., Moal, I. H., and Deane, C. M. (2024). Addressing the antibody germline bias and its effect on language models for improved antibody design. *bioRxiv*.

Prihoda, D., Maamary, J., Waight, A., Juan, V., Fayadat-Dilman, L., Svozil, D., and Bitton, D. A. (2022). Biophi: A platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning. *mAbs*, 14(1):2020203. PMID: 35133949.

Purshouse, R. and Fleming, P. (2003). Evolutionary many-objective optimisation: an exploratory analysis. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 3, page 2066–2073. IEEE.

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural Networks*.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

Ren, M., He, Z., and Zhang, H. (2024). Multi-objective antibody design with constrained preference optimization.

Ren, Z., Li, J., Ding, F., Zhou, Y., Ma, J., and Peng, J. (2022). Proximal exploration for model-guided protein sequence design. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18520–18536. PMLR.

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

Shuai, R. W., Ruffolo, J. A., and Gray, J. J. (2023). IgLM: Infilling language modeling for antibody sequence design. *Cell Systems*, 14(11):979–989.e4.

Sinai, S., Wang, R., Whatley, A., Slocum, S., Locane, E., and Kelsic, E. D. (2020). AdaLead: A simple and robust adaptive greedy search algorithm for sequence design.

Stanton, S., Maddox, W., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. (2022). Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *Proceedings of the 39th International Conference on Machine Learning*, pages 20459–20478. PMLR. ISSN: 2640-3498.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020.

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and De Freitas, N. (2013). Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 1778–1784. AAAI Press.

Widatalla, T., Rafailov, R., and Hie, B. (2024). Aligning protein generative models with experimental fitness via direct preference optimization.

Yang, J., Chu, W., Khalil, D., Astudillo, R., Wittmann, B. J., Arnold, F. H., and Yue, Y. (2025). Steering generative models with experimental data for protein fitness optimization.

Yang, K. K., Wu, Z., and Arnold, F. H. (2019). Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694.

Zhou, X., Xue, D., Chen, R., Zheng, Z., Wang, L., and Gu, Q. (2024). Antigen-specific antibody design via direct energy-based preference optimization. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 120861–120891. Curran Associates, Inc.

A Details about Methods

A.1 Sequence Encodings

One-hot Each amino acid gets encoded as a one-hot vector which get concatenated to encode a sequence of amino acids.

BLOSUM We follow (Oglic and Gärtner, 2018, 2019; Gessner et al., 2024) and use the eigendecomposition UDU^T of the block-substitution matrix (BLOSUM) to construct embedding vectors $U|D|^{1/2}$. BLOSUM (Henikoff and Henikoff, 1992) is an indefinite matrix that quantifies similarities between amino acids by recording the effect of their substitution in proteins.

A.2 Genetic Optimizer

To generate a new generation in the genetic optimizer, we repeatedly apply:

Tournament selection Sample a subset of the sequences from the previous generation and retain the best-scoring sequence.

Single-point crossover Having sampled two parents via tournament selection, we create two offsprings by randomly cutting both parental sequences at a sampled position and swapping the remaining sequence after this position. We apply crossover with a rate of 0.7, otherwise the parents make it to the next step.

Mutation We then apply random mutations to amino acids in the sequence. We use a per-position mutation probability of 0.1.

This procedure is repeated until the new generation has the desired size. If not stated otherwise, we use an initial population of size 50, and 50 sequences per generation over 20 generations. The score used is the value of the acquisition function at that point, evaluated with the surrogate model.

Not only is the GA a natural choice for sequence optimization, it also permits easy incorporation of constraints. We can easily restrict the positions that we want to permit mutation in, and restrict the allowed mutations in each location based on expert knowledge. Furthermore we incorporate liability filtering to prevent introducing glycosylation sites and to exclude sequence motifs that are known to affect stability or other properties of the antibody.

The GA has to be modified for the batch BO version, where the acquisition function is jointly defined over a batch of sequences, i.e. $\alpha_q : \mathcal{S}^q \to \mathbb{R}$. Hence, the GA no longer evolves individual sequences, but batches of them. We introduce an additional batch-crossover operation that generates offspring batches from two parental batches by performing single-point crossover between sequences in the other batch and swapping sequences between batches with a batch crossover rate of 0.7.

Instead of the acquisition function, we can directly interface the objective function as a fitness function in the GA. This makes the GA an obvious baseline to compare to. In the case of multi-objective optimization, we employ a sum of normalized scores as the fitness function.

B Experimental Details

In all experiments, we generate 100 initial sequences with 2 maximum mutations for each of 10 different random seeds per method. All methods were allowed up to 1000 oracle calls to evaluate sequences. Batch acquisition functions had a batch size of 4, so were run for 250 iterations; sequential EHVI was run for 1000 iterations. All GAs (baselines and within BOAT) scored 50 sequences per generation over 20 generations. We used one-hot encoding. GA settings were as in Section 3.3 for both the GA baseline and within the Bayesian optimization loop, except that the mutation probability was set as 0.15 for all GAs except BOAT runs with the qEHVI acquisition. This was to promote diversity due to the large number of iterations that other algorithms were run for.

NSGA-II is a GA specifically tailored for multi-objective optimization (Deb et al., 2002). NSGA-II maintains population diversity by mutating solutions along the Pareto frontier and using crowding distance measures to promote diversity along the Pareto front, though performance degrades with increasing objectives (Purshouse and Fleming, 2003). We use the version implemented in PyMoo (Blank and Deb, 2020). For NSGA-II, we use the version implemented in PyMoo (Blank and Deb, 2020) with custom mutation and crossover functions appropriate for sequences.

The GA comparison in this experimental setup is feasible as the objectives used in this section are fast to evaluate. Running the GA within the inner-loop of the Bayesian optimization takes less than one second in most cases. However, we reiterate that GAs are generally not suitable for tasks when objective functions require expensive experimental evaluation or lengthy simulations, highlighting a key advantage of BO.

Limiting the search space to a maximum of 5 mutations per CDR allows us to brute-force the computation of the complete Pareto front. We emphasize that direct access to the ‘ground truth’ Pareto front is rarely available, as the design space is typically vast and exhaustive evaluation using computational oracles is prohibitively expensive.

C Additional Results

C.1 2 objectives

Figure 3 shows the hypervolume evolution for the cross-reactivity experiment, in which we optimize CDRs separately with respect to two affinity-related objectives. We allow up to 5 mutations on each CDR.
Figure 4 show the Pareto front found by different methods for all seeds that have been average over in Figure 3. We can see that while the GA and NSGA-II sometimes find solutions close to the ground truth Pareto front, the quality of their outcome strongly depends on the initial seed, which is less so for all BOAT variants.

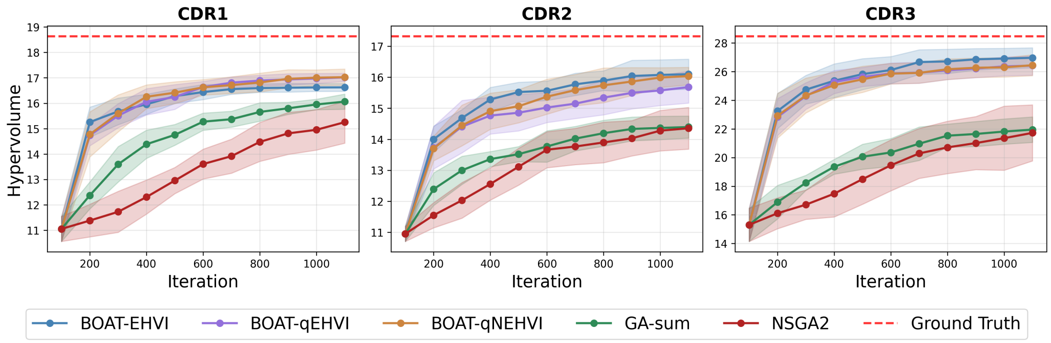
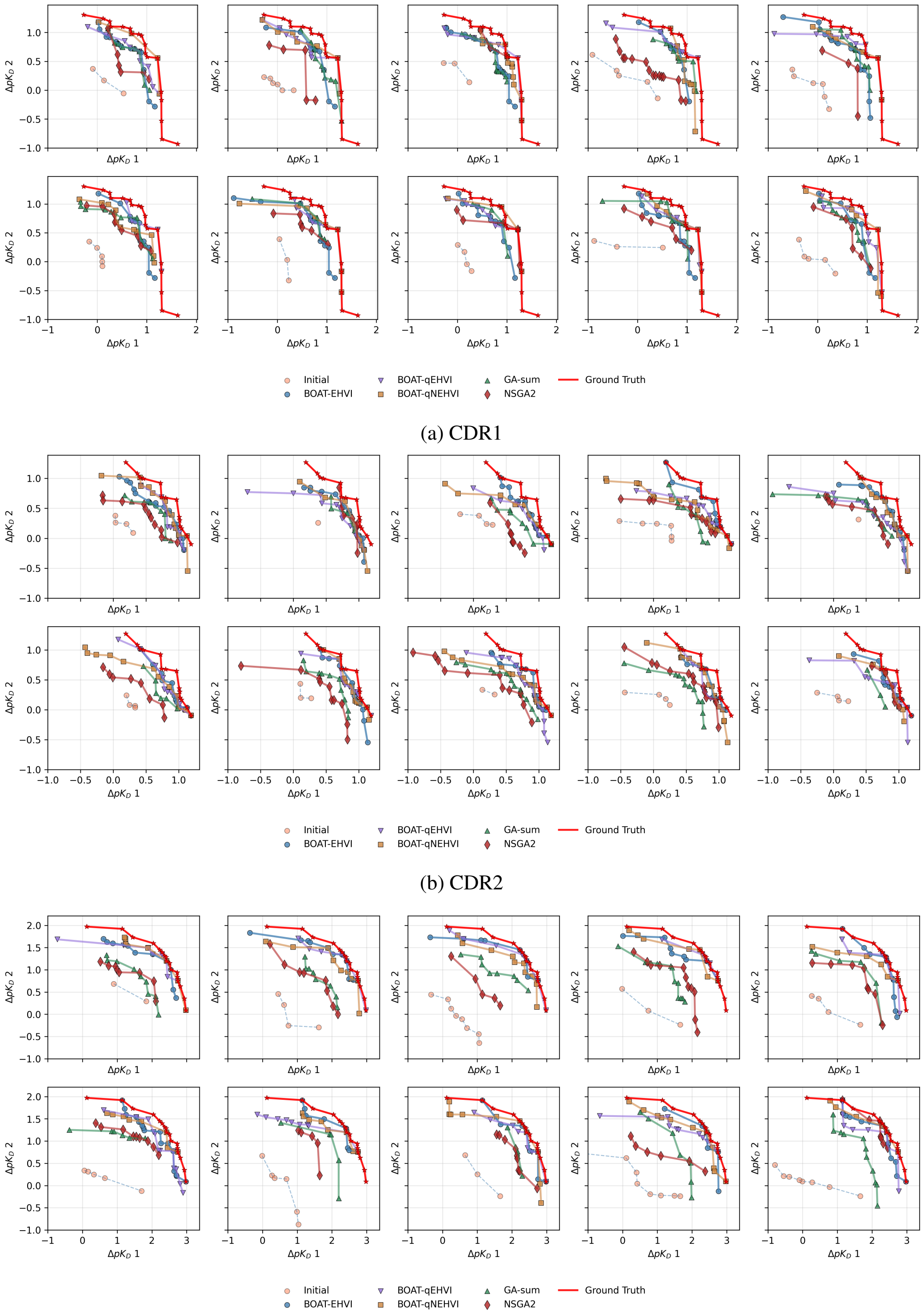


Figure 3: Hypervolume evolution for 2-objective CDR optimization. Results averaged over 10 seeds with standard error.



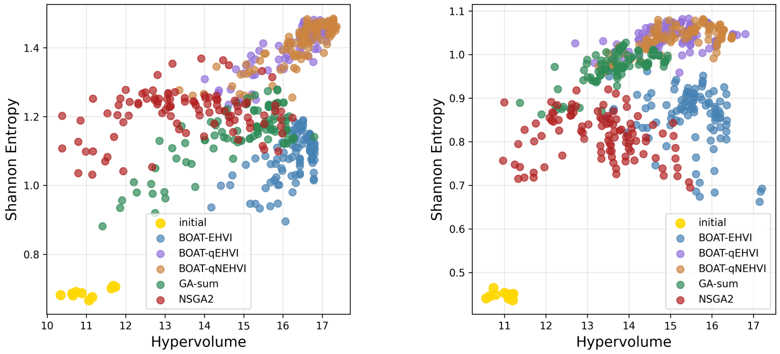
(a) CDR1

(b) CDR2

(c) CDR3

Figure 4: Plot comparing Pareto front by seed across different conditions for each CDR. All seeds are shown.

Figure 5 and Figure 6 display sequence diversity for all CDRs as in Figure 1b and Figure 1c. The BOAT variants consistently find sequences with a larger hypervolume and greater sequence diversity.



(a) CDR1

(b) CDR2

Figure 5: Scatter plots of hypervolume versus Shannon entropy for CDR1 and CDR2 optimization. Each point represents the diversity and multi-objective performance of a population at a given optimization step, showing results for all seeds, methods, and every 100 iterations. Initial solutions are highlighted in gold.

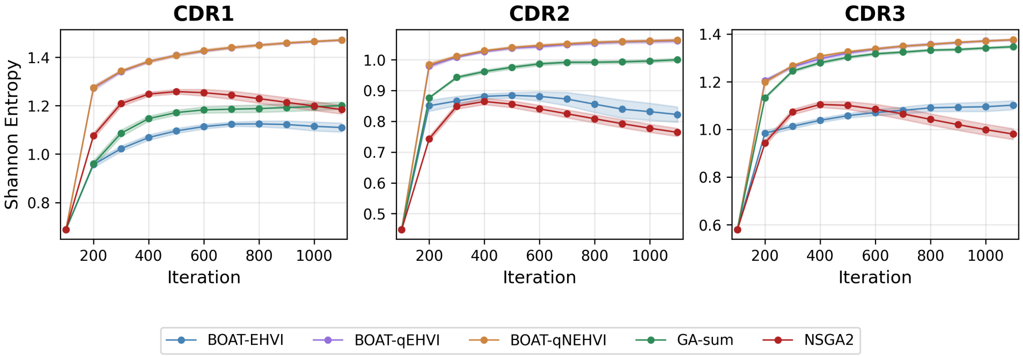


Figure 6: Evolution of Shannon entropy over optimization iterations for CDR1, CDR2, and CDR3 regions. Results averaged over 10 seeds with standard error bands, showing how population diversity changes throughout the optimization process across all three CDR regions.

C.2 3 objectives

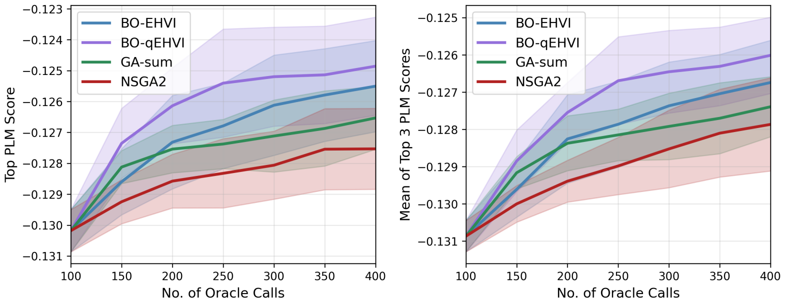


Figure 7: PLM evolution for the first 300 generations of 3-objective CDR optimization, with the best PLM score of all generated sequences and the mean score of the top 3 PLM scores recorded. Results averaged over 10 seeds with standard error. Note that the x-axis starts at 100 to omit the initial sequences.

C.3 4 objectives

We added a fourth objective: sequence likelihood from the ESM protein language model. Due to the computational expense of querying this model, we only compared batch BOAT to our GA baselines. It was infeasible to score all our ground truth sequences.

Figure 8 shows the hypervolume evolution over optimization iterations, with BOAT outperforming both baselines for all CDRs. BOAT maintains its effectiveness even in higher-dimensional objective spaces where traditional multi-objective optimization with GAs becomes increasingly challenging.

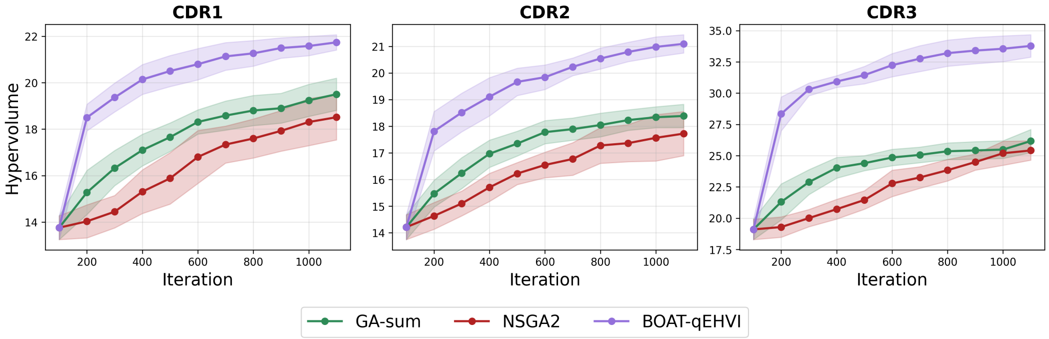


Figure 8: Hypervolume evolution for 4-objective CDR optimization. Results averaged over 10 seeds with standard error.