BOAT: Navigating the Sea of In Silico Predictors for Antibody Design via Multi-Objective Bayesian Optimization

Jackie Rao[1] **Ferran Gonzalez Hernandez**[2] **Leon Gerard**[2] **Alexandra Gessner**[2]
[1]MRC Biostatistics Unit, University of Cambridge, UK
[2]Centre for AI, 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 and newer generative learning approaches, we demonstrate competitive performance with state-of-the-art methods for multi-objective protein optimization. We identify clear regimes where surrogate-driven optimization outperforms expensive generative approaches and establish practical limits imposed by sequence dimensionality and oracle costs.

1 INTRODUCTION

Lead optimization lies at the heart of therapeutic antibody development, where the goal is to advance promising candidates into clinically viable drugs. In this process, candidates are systematically improved to meet multiple, often competing, criteria such as binding affinity, manufacturability, biophysical stability, and immunogenicity. The specific combination of properties targeted can vary significantly between campaigns, with some requiring cross-species reactivity to enable testing human therapeutics in animal models, while others prioritize developability or other traits. Optimization efforts typically focus on the complementarity-determining regions (CDRs) of the antibody, the variable loops responsible for antigen binding (Sela-Culang et al., 2013). Heavy chain CDRs are often prioritized, particularly CDR-H3, which exhibits the greatest diversity and contributes most significantly to binding (Xu and Davis, 2000). As the number of required properties grows, the complexity of searching for optimal antibody sequences quickly outpaces what can be achieved through traditional trial-and-error or single-target screening methods. The exponential growth in sequence and property space creates a pressing need for systematic strategies that can efficiently navigate these multidimensional landscapes.

Modern *in silico* approaches have emerged as indispensable tools in addressing these challenges, allowing scientists to rapidly predict and evaluate protein features before experimental validation. Machine learning-based property predictors and physics-based simulations offer the potential to assess vast libraries of antibody variants at a much lower cost than experimental validation. Still, the computational resources required for tasks such as structure prediction or simulating relative binding free energies are substantial. Hence, a systematic approach is needed to decide which candidates to score. Furthermore, integrating these predictive models for multiple objectives presents methodological hurdles: conflicting property requirements may restrict sequence innovation, and poor predictive power complicates decision-making. Therefore, methodologies capable of jointly optimizing multiple objectives while quantifying uncertainty are

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

vital for designing new experiments and steering antibody engineering towards the most promising leads.

We address these challenges with BOAT (**B**ayesian **O**ptimization for **A**ntibody **T**raits), a versatile multi-objective Bayesian optimization framework for antibody sequences, illustrated in Figure 1. BOAT supports easy interfacing of arbitrary *in silico* predictors, and allows for either full sequence optimization or region-specific optimization. This allows users leverage appropriate scoring functions depending on particular requirements posed in lead optimization campaigns. BOAT has been constructed as the inner loop in a single sequence design step of an outer "wetlab loop". While data from previous wetlab experiments may be used to inform oracles, BOAT remains agnostic of experimental data. Thus, it remains up to the user to validate their selected oracles prior to optimizing them with BOAT. While we focus here on antibodies, the approach extends naturally to other therapeutic proteins and small molecules. In this work, we consider models for binding affinity, humanness evaluation, as well as structure prediction for joint optimization. Humanness prediction assesses how closely an engineered antibody sequence resembles naturally occurring human antibodies—a critical property for therapeutic development, as higher humanness typically reduces immunogenicity and improves safety profiles.

Key Contributions

- We construct a light-weight 'plug-and-play' Bayesian multi-objective optimization framework to optimize antibody lead candidates against computationally predicted properties of interest. Code can be found at `https://github.com/AstraZeneca/boat`.
- We perform rigorous benchmarking of Bayesian-based optimization with surrogate models versus genetic and generative baselines, quantifying oracle efficiency, diversity and Pareto front quality.
- We demonstrate that our method efficiently explores the Pareto front where the combinatorial ground truth is available. This approach enables the systematic identification of Pareto optimal candidates, allowing for the selection of antibodies that represent balanced trade-offs between multiple objectives according to specific priorities.
- We deliver guidelines for the selection of approaches for experimental design in antibody engineering, grounded in systematic scaling and integration of both inexpensive and computationally demanding oracles.

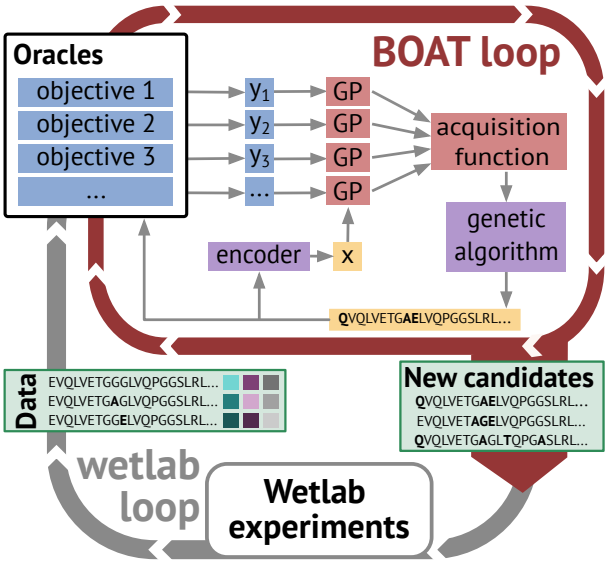


Figure 1: Illustration of the BOAT loop for lead optimization (LO). LO campaigns typically consist of a few rounds of experimental testing of the order of a hundred sequences in the wetlab, where previous experiments inform the sequence design for the next iteration of the wetlab loop. The BOAT loop can be thought of as an inner loop in a single sequence design step of the wetlab loop, leveraging sophisticated tools used for computationally assessing sequences. The multi-objective optimization loop eliminates the common practice of generating a large pool of sequences and filtering them down by sequentially passing them through the different oracles, which is computationally inefficient and unaware of Pareto trade-offs. The oracles used depend on the requirements of the LO campaign, and can be exchanged flexibly. See Section 2 for details on the internal building blocks of BOAT.

2 MATERIALS AND METHODS

2.1 Multi-Objective Bayesian Optimization

Bayesian Optimization (BO) provides a sample-efficient framework for global optimization by maintaining a probabilistic surrogate model of the objective function and using acquisition functions to guide the search. See Frazier (2018); Garnett (2023) for a detailed introduction to BO. Generally, BO seeks to find the maximum of a function $f : \mathcal{X} \to \mathbb{R}$ in as few evaluations as possible, where f is often expensive to evaluate and lacks structure (e.g., closed-form gradients) that would make it amenable to direct optimization methods.

Given a dataset of previous (potentially noisy) evaluations $\mathcal{D}_t = (\mathbf{x}_i, y_i)_{i=1,\dots,t}$, a probabilistic surrogate

model $p(f|\mathcal{D}_t)$ is fit to this dataset which captures the current belief about the unknown objective function f, where $y_i = f(\mathbf{x}_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. A Gaussian process (GP) (Williams and Rasmussen, 2006) is the most commonly used surrogate model, enabling non-parametric regression and providing both mean and uncertainty estimates at each input point $\mathbf{x}$. GPs are popular due to their closed-form posteriors, flexibility in kernel choice for encoding different data structures and inductive biases, and strong performance in data-scarce regimes.

An acquisition function $\alpha(\mathbf{x}|\mathcal{D}_t)$ quantifies the utility of evaluating f at each candidate input point, given predictions from the surrogate model, balancing exploitation of regions with high predictive performance against exploration of uncertain and under-explored areas. The next evaluation point is selected by optimizing this acquisition function: $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}|\mathcal{D}_t)$. After evaluating $y_{t+1} = f(\mathbf{x}_{t+1})$, the dataset is augmented and the process repeats until a stopping criterion is met or the evaluation budget is exhausted. For single-objective optimization, we use Log Expected Improvement (LogEI) (Ament et al., 2023), and for multi-objective optimization, 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) and whose modular design enables straightforward extension to additional acquisition functions.

2.2 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. 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 the following sequence encodings,

One-hot Each amino acid gets encoded as a one-hot vector; sequences are encoded as a concatenation of one-hot encoded amino acids.

Bag of amino acids To include sequence motifs beyond individual amino acids, we encode matching n-grams (with n=5), similar to the bag of words embedding.

BLOSUM We follow (Oglic and Gärtner, 2018, 2019; Gessner et al., 2024) and use the eigendecomposition UDU^T of the block-substitution matrix (BLOSUM) with similarity 45 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.

AbLang-2 AbLang-2 is an antibody-specific protein language model providing embeddings of antibody sequences taking into account learned context across the sequence (Olsen et al., 2024).

The embedding space is typically quite large for all considered embeddings. For example, both one-hot and BLOSUM give rise to sequence embeddings of size sequence length $\times$ number of amino acids (i.e., 20). We employ a Gaussian process model that has been designed for this kind of high-dimensional problem, using the Tanimoto kernel (Ralaivola et al., 2005)

$$k_{\text{Tanimoto}}(\mathbf{x}, \mathbf{x}') = \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - \langle \mathbf{x}, \mathbf{x}' \rangle}. \quad (1)$$

The Tanimoto similarity was initially used to compare binary molecular fingerprints of small molecules, but has been extended for more general molecular embeddings that lie in $\mathbb{R}^d$.

2.3 Genetic Optimizer

While some embeddings, such as one-hot and BLOSUM, admit an explicit reconstruction of amino acid sequences, they still represent fundamentally discrete objects. Although the embeddings are vector-valued, each dimension corresponds to categorical amino-acid choice, and most points in this vector space do not correspond to valid sequences. While the acquisition function is differentiable in principle, a gradient-based optimizer would move through arbitrary real-valued vectors; projecting these vectors back to the nearest valid discrete sequence would result in large, semantically meaningless jumps. We instead use a genetic algorithm (GA), a discrete optimization method, to generate 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 a local minimum of the acquisition function. To generate a new generation, we repeatedly apply:

Tournament selection Sample a subset of 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 either 0.1 or 0.15.

This procedure is repeated until the new generation has the desired size. The parameters have been chosen from an initial tuning phase. 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 the introduction of glycosylation sites and to exclude sequence motifs that are known to affect stability or other properties of the antibody.

The GA is 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 a 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 multi-objective optimization, we employ a sum of normalized scores as the fitness function.

2.4 Oracles

Affinity predictor We train a neural network predictor on experimental affinity data for each considered antibody-antigen pair to predict relative improvement of affinity over the parental. To handle 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 to predict the delta in binding affinity with respect to a reference sequence (e.g., parental sequence).

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).

Structure prediction Structure prediction tools do not inherently predict antibody properties that can be measured in the laboratory. However, they provide scores that represent the confidence of the model about the predicted structure. We use the interface predicted TM score (ipTM) from Boltz-2 (Passaro et al., 2025) to score model confidence at the interface between the antibody and the antigen, a metric with potential correlation to binding signal (Zambaldi et al., 2024). With a runtime in the order of minutes per sequence on a GPU, this is the slowest objective we are considering.

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. Building purely on oracles, BOAT does not directly address the issue of oracle quality - it is up to the user to decide which oracles to use for a particular campaign. This design choice reflects the fact that experimental measurements are costly and slow to obtain, and that *in silico* predictors are imperfect but often serve as the best available proxies during lead optimization. We can leverage well-studied and sophisticated predictors instead of relying on predictors trained purely on small experimental datasets. In principle, the framework supports weighting oracles according to user preference - for example, the Expected Weighted Hypervolume Improvement criterion extends EHVI to accommodate weighted objectives (Feliot et al., 2018).

3 RELATED WORK

Traditional and Evolutionary Baselines Genetic and evolutionary optimizers have long been used for optimizing black-box functions in discrete spaces; see Katoch et al. (2021) for a recent review. These algorithms gradually evolve a solution through random

mutation, crossover and selection without gradients. Multi-objective extensions like NSGA-II (Deb et al., 2002) optimize for diverse Pareto-optimal solutions.

Traditional approaches are often inefficient in protein design given the high-dimensional sequence space (Turner et al., 2021). This has motivated specialized protein evolutionary algorithms that incorporate domain-specific knowledge, such as AdaLead (Sinai et al., 2020) and PEX (Ren et al., 2022). Some methods use neural networks or language models to guide mutation selection (Nigam et al., 2019; Yang et al., 2019; Nana Teukam et al., 2025).

Generative Models and Reinforcement Learning (RL) Recent advances in generative modeling have enabled the synthesis of novel, plausible antibody sequences by learning from large corpora of protein data. Key approaches include transformer-based protein language models trained using either masked language modeling or next-token prediction objectives (in autoregressive models) (Rives et al., 2021; Ferruz et al., 2022; Nijkamp et al., 2023). Most autoregressive language models are limited to generating full sequences, but there exist extensions for conditional infilling to redesign specific regions like CDRs (Shuai et al., 2023; Melnyk et al., 2023). Diffusion models represent another promising direction (Ho et al., 2020); see He et al. (2025) for a recent review. These models are increasingly fine-tuned to steer generation towards sequences or edits which respect developability constraints and optimize certain objectives (Goel et al., 2024; Yang et al., 2025). RL offers a flexible framework to align pre-trained generative models with specific experimental objectives, often employing policy gradient methods such as REINFORCE to optimize the generation towards specific properties. Both proximal policy optimization (PPO) (Angermueller et al., 2019; Lee et al., 2025a) and Direct Preference Optimization (DPO) (Widatalla et al., 2024) have been applied to single-objective protein design.

Bayesian Optimization BO uses uncertainty-calibrated surrogates and acquisition functions for sample-efficient optimization of expensive black-box objectives. Rather than learning to generate sequences directly, BO methods iteratively propose candidates by balancing exploration of uncertain regions with exploitation of high-performing areas in the search space, and evaluate proposals using a predictive oracle or real experimental data. However, most current BO frameworks target single objectives or specific architectures, limiting applicability to multi-objective antibody design; (González-Duque et al., 2024) provides a recent survey of BO methods for antibody design. A key challenge is the high-dimensional nature of discrete sequence space (Wang et al., 2016).

Latent Space Bayesian Optimization employs a BO framework within a continuous latent space to search for optimal sequences. While many approaches use a Variational Autoencoder (VAE) pretrained on a large dataset (Gómez-Bombarelli et al., 2018; Tripp et al., 2020; Notin et al., 2021; Lee et al., 2023; Moss et al., 2025), recent advances exploit the robust feature learning capabilities of Denoising Autoencoders (DAEs) (Maus et al., 2022; Stanton et al., 2022; Gruver et al., 2023). However, ensuring decoded sequences remain plausible is challenging. Lee et al. (2025b) approach this in single-objective sequence optimization using autoregressive normalizing flows to eliminate the reconstruction gap. These methods also require a very large training dataset.

Other approaches operate directly in sequence space using specialized kernels and discrete optimization methods. BOSS (Moss et al., 2020) employs string kernels with genetic algorithms for acquisition optimization, while AntBO (Khan et al., 2022) uses a Transformed Overlap Kernel (TK) and a deep ProteinBERT (Brandes et al., 2022) kernel for CDRH3 optimization. AntBO additionally employs trust-region-based search restrictions (Eriksson et al., 2019), which can help the search in high dimensions (Zhang et al., 2021). Recent work has also explored hybrid approaches that combine generative modeling with BO principles. CloneBO (Amin et al., 2024) combines language models trained on clonal families with Thompson sampling for biologically-informed optimization.

Multi-Objective Optimization Few approaches directly consider multi-objective optimization, where the hypervolume indicator is typically used to quantify the quality of the Pareto front. Ren et al. (2025) incorporates multiple objectives as constraints in an RL framework, while other approaches use gradient-based optimizers requiring differentiable predictors or lengthy computation (Emami et al., 2023; Luo et al., 2025). LaMBO and LaMBO-2 (Stanton et al., 2022; Gruver et al., 2023) extend a BO framework with DAEs and generative infilling to multiple objectives, using expected hypervolume improvement (EHVI) as an acquisition function. ALLM-Ab (Furui and Ohue, 2025) uses a fine-tuned protein language model in an active learning framework where sequences are selected based on hypervolume maximization. Our work represents the first multi-objective BO framework for optimizing black-box in-silico predictors directly in discrete sequence space.

4 EXPERIMENTS

We evaluate our multi-objective Bayesian optimization framework across three experimental settings: the optimization of a single-domain antibody with a focus

on cross-reactivity, a benchmark comparison against LaMBO-2, and additional studies examining the impact of key design choices in the Appendix.

4.1 Cross-reactivity of a V_{HH}

We ran BOAT on a therapeutic nanobody (V_{HH}), a single-domain antibody derived from the heavy chain variable domain, 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 single-point mutations that we have experimental data for. This way we prevent reliance on an oracle that has not observed certain mutations, and because BOAT and all baselines use the same dictionary, it does not introduce bias into the comparative evaluation. This setting reduces the size of the search space and allows us to brute-force the computation of the complete 'ground truth' Pareto front, defined as the Pareto front induced by exhaustively enumerated oracle scores, for up to 3 objectives.

Our goal is to evaluate whether BOAT can efficiently recover the Pareto front as defined by the *in silico* predictors and explore the sequence space to optimize the predictor values; the true experimental landscape is not available for these antibody systems. 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. 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 2.4 that were trained on the experimentally measured affinities for both antigens of 340 single-point and 26 quadruple mutations. We add the humanness score (third) and a PLM log-likelihood (fourth - where computing the ground truth was intractable) as additional objectives (cf. Section 2.4).

We benchmarked against two GA baselines. Our first GA baseline was set up as a standard GA (described in Section 2.3) which optimizes a normalized sum of the objectives. Our second GA baseline is NSGA-II (Deb et al., 2002), a GA specifically tailored for multi-objective optimization. NSGA-II maintains population diversity by mutating solutions along the Pareto frontier, though performance degrades with increasing objectives (Purshouse and Fleming, 2003).

We evaluated our methods by comparing their discovered Pareto fronts to the ground truth where available, and track how the hypervolume evolves over oracle calls. By default, BOAT computes the reference point at the start of optimization as the minimum of the initially scored sequences minus 10%. For fair and consistent comparison of hypervolumes across different initializations, we fixed the hypervolume reference point across all experiments. We set it to $[-3, -3, 0, -1]$ for the two affinity predictors, humanness score, and the PLM log-likelihood respectively, using prior knowledge of the oracle score ranges. 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. See Appendix A for further experimental details. We note that the ground truth hypervolume and search space is much higher for CDR3. It is easier to destabilise binding for CDR3, and CDR3 is known to be the most important for antigen recognition (Xu and Davis, 2000). We focus ablations on CDR3, comparing encodings and batch sizes for batch BO in Appendix C and E, respectively.

4.1.1 Hypervolume evolution

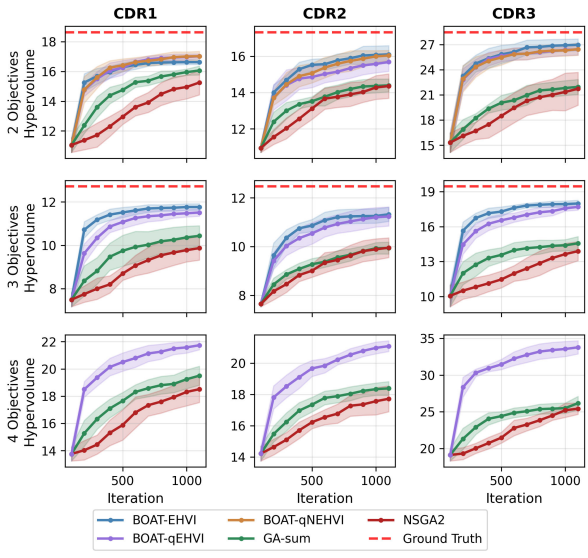


Figure 2: Hypervolume vs. iterations (mean ± s.e., 10 seeds) for CDR1–3 under 2, 3, and 4 objectives. BOAT variants (q indicates batch acquisition functions) outperforms GA-sum and NSGA-II. Red dashed lines show ground-truth hypervolume where available.

Figure 2 shows the hypervolume evolution as a function of oracle calls for all methods across CDR1–CDR3 and across 2, 3, and 4 objectives. BOAT variants con-

sistently reach higher hypervolume earlier and achieve larger final hypervolume than GA baselines, and maintains its effectiveness even in higher-dimensional objective spaces. Batch acquisition (qEHVI, qNEHVI) favours broader exploration early with a lower hypervolume, but there was no significant difference in the final hypervolume found between acquisition functions. NSGA-II underperforms progressively as the number of objectives increases, consistent with prior reports.

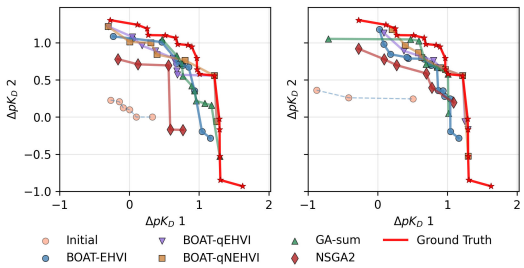
qNEHVI became much slower when increasing the number of objectives; while two-objective qNEHVI evaluations complete in seconds, three objectives requires several minutes per BO step ($>$ 100 times slower). This rapid degradation is consistent with findings in Daulton et al. (2021). As qNEHVI did not outperform qEHVI in the 2-objective setting, we did not use this for more objectives. For the 4-objective setting, querying the PLM is the bottleneck, so we report batch BOAT versus GA baselines only.

4.1.2 Validation against ground truth, sequence diversity, and PLM

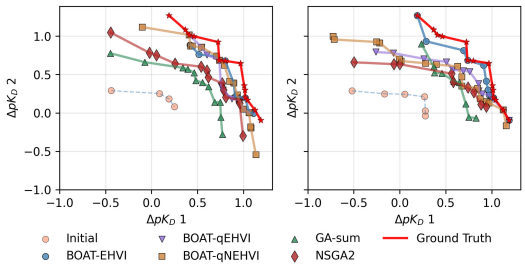
In 2 dimensions, we can visualize the discovered Pareto front against the 'ground truth' Pareto front derived from exhaustive evaluation. A subset of these plots is visualized in Figure 3, displaying the seed with the highest hypervolume among all GA methods and the seed with the highest hypervolume among all BOAT variants for that CDR. Plots for all seeds and also plots for earlier points in all the models are in Appendix D. BOAT traces fronts close to the true frontier and often recovers true Pareto-optimal sequences even in CDR3's 63M-sequence space.

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 in the 3-objective setting. Figure 4 reveals that BOAT produces sequences with slightly higher PLM scores early in the optimization process without explicitly optimizing for this. Multi-objective BO naturally favours sequences with better biological fitness, even when all methods operate under identical mutation constraints.

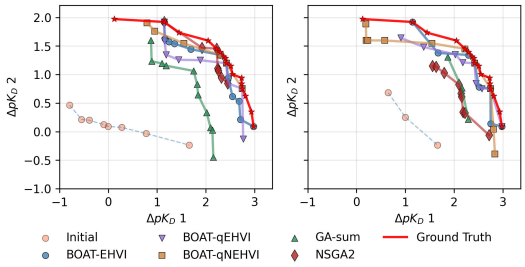
To assess the diversity of solutions discovered by each method, we computed the average Shannon entropy for all generated sequences for every 100 generated sequences, for all methods in the 2-objective setting. A visualization of the results for CDR3 can be seen in Figure 5a, comparing Shannon entropy to hypervolume and iteration, with additional figures for other CDRs in Appendix B. Batch acquisition methods (BOAT-qEHVI, BOAT-qNEHVI) achieve the optimal combination of both high hypervolume performance



(a) CDR1



(b) CDR2



(c) CDR3

Figure 3: Discovered vs. ground truth Pareto fronts for 2-objective CDR optimization with 5 mutations. Each panel shows the best-performing seed for GA methods (left) and BOAT (right). We can see that even the most successful GA runs fall behind BOAT.

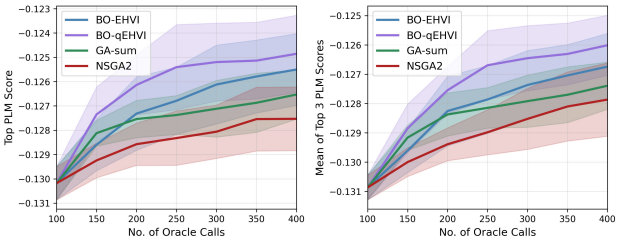
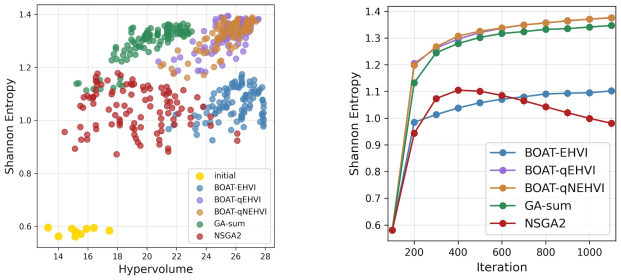


Figure 4: PLM evolution for the first 300 generations of 3-objective CDR3 optimization, with the best PLM score of all generated sequences and the mean score of the top 3 PLM scores recorded. Results averaged over 10 seeds with standard error. Note that the x-axis starts at 100 to omit the initial sequences.

and high sequence diversity. The larger diversity likely stems from the batch acquisition's inherent mechanism of selecting multiple diverse candidates simulta-

neously, naturally promoting exploration of different regions of the sequence space. BOAT-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. Figure 5b shows that BOAT successfully maintains sequence diversity throughout the algorithm.



(a) Diversity vs. HV (b) Evolution of diversity

Figure 5: (a) Scatter plot of hypervolume versus Shannon entropy for all seeds, methods, and every 100 iterations for CDR3. Initial solutions are highlighted in gold. Each point represents the diversity and multi-objective performance of a population at a given optimization step. (b) Evolution of Shannon entropy over optimization iterations for CDR3. Results averaged over 10 seeds with standard error bands.

4.1.3 Structure prediction oracle

We further consider a 3-objective setup with two affinity predictors and Boltz-2 for structure prediction, where we considered ipTM the score of interest to optimize; further details are in Appendix A.2. Boltz-2 metrics are challenging oracles that rely on a powerful 3D antibody-antigen representation from the AlphaFold Pairformer. Unsurprisingly, the Tanimoto model with BLOSUM encoding struggles to capture this structural complexity. The GA interestingly achieves comparable performance to Bayesian optimization methods via semi-random mutations in this scenario, cf. Figure 6. Remarkably, NSGA-II does consistently worse here.

4.2 4-4-20 scFv Antibody

4.2.1 Dataset and Experimental Setup

We now run a comparison of multi-objective optimization between BOAT and LaMBO-2 (Gruver et al., 2023). LaMBO-2 requires a corpus of training data to be used as it trains its own predictive model f^*, rather than allowing for the use of any external oracle in its optimization loop. We use a public dataset from Adams et al. (2016), which is featured in the Fitness Landscape of Antibodies (FLAb) benchmark (Chungyoun et al., 2024). This consists of 10K+ affinity and expression measurements derived from mutating CDR1 and CDR3 regions of the 4-4-20 scFv antibody (Boder and Wittrup, 1997). This dataset contains between 3 and 18 repeated measurements for each antibody, so we take a mean of the repeats and retain only sequences with both affinity and expression measurements, resulting in 2807 total sequences. We extracted the predictive model f^* from LaMBO-2's trained discriminative head as the black-box oracles used in BOAT for prediction of affinity and expression, enabling a fair comparison between the methods.

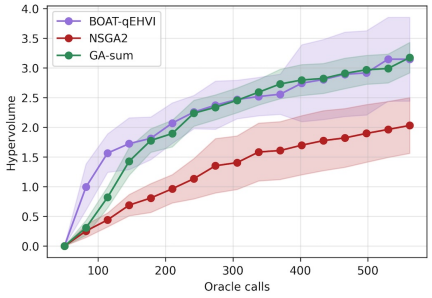


Figure 6: Hypervolume evolution for 3-objective CDR optimization. Initial HV has been subtracted. Results averaged over 5 seeds with standard error.

We generated 256 sequences from each method; further details can be found in Appendix G.1. We limit mutations to CDR1 and CDR3 and allow BOAT to introduce up to 8 mutations, a constraint used in a lead optimization study in (Gruver et al., 2023). This allows for more mutational freedom than previously as we edit two CDRs.

4.2.2 Results

We see from Figure 7 that the hypervolume found by BOAT is generally larger than for LaMBO-2 throughout optimization. The predicted scores of the generated sequences indicate that BOAT is often able to push further toward Pareto-optimality than LaMBO-2. However, LaMBO-2's saliency-guided editing mechanism encourages mutations to amino acids observed in the training data, potentially generating more biologically realistic sequences backed with training data, whereas BOAT does not impose biological priors. This may enable exploration of less common, yet beneficial mutations.

To constrain BOAT to more realistic sequences, we added ESM-2 likelihood as a third oracle, but only compare the hypervolume given by the affinity and expression predictions in Figure 7. This counteracts the tendency of BOAT to discover sequences with implausible predictive values relative to the training distribution. These artifacts arise due to overfitting and mislead the optimization process toward sequences with artificially inflated predicted performance. Given the limited and imbalanced training data, such behaviour is not unexpected but reflects common issues in *in silico* lead optimization. Appendix G.2 contains further insights and discussion about the respective merits of LaMBO-2 and BOAT.

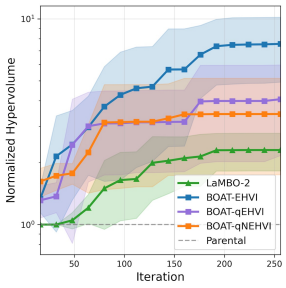


Figure 7: Comparing relative hypervolume versus number of generated sequences between BOAT and LaMBO-2. The reference hypervolume of 1 has been assigned to the parental sequence.

This comparison has inherent limitations that merit discussion. LaMBO-2's architecture tightly couples a generative diffusion model with a discriminative head through shared layers, making it impossible to substitute external oracles for evaluation. LaMBO-2 is elegant in that it condenses the entire computational antibody design into a single procedure without human intervention. This is powerful for designing sequences purely from experimental measurements of candidate traits. However, the human-in-the-loop design approach taken with BOAT is preferential to leverage both expert knowledge and state-of-the-art external predictors for diverse properties, some of which may not be directly measured in wet-lab experiments. Our plug-and-play approach addresses this limitation by treating predictors as modular components that can be easily swapped or added, while achieving comparable or superior performance in identifying high-performing sequences.

5 DISCUSSION

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 Pareto trade-offs between multiple objectives. BOAT allows users to interface arbitrary tools for antibody property prediction, enabling the joint optimization of existing state-of-the-art *in silico* oracles. BOAT operates directly in sequence space, does not require any pre-training, and performs competitively with only a few initial scored sequences. The success on two antibody candidates highlights a critical gap in current protein design methodology: most existing approaches optimize single objectives or require extensive pre-training, yet real-world antibody development demands simultaneous optimization of multiple properties with small experimental budget and long timescales. BOAT bridges this gap through its modular design, allowing users to easily interface external oracles - which are not limited to those included in this work. This flexibility makes it practical for real-world antibody design.

While being simple and versatile, BOAT has important limitations and potential for future research. We observed in Section 4.1.3 that the oracle's complexity can compromise the performance of the BOAT's surrogate model. A step ahead would be to leverage more tailored surrogates for antibody modelling, either through protein-specific kernels (Groth et al., 2024) or encodings that capture antibody structure (Malherbe and Ucar, 2024). Section 4.2 revealed that biological priors can prevent the generation of out-of-distribution sequences, and could be straightforwardly included in the GA by sampling from a PLM likelihood instead of uniformly. Other areas of research can look into other acquisition functions; trust region Bayesian optimization (Eriksson et al., 2019) has seen success in other high-dimensional regimes. While BOAT still explored sequence space well with 4 objectives, it is known that GPs perform poorly in very high dimensions (Binois and Wycoff, 2022). While it is likely that some filtering by certain properties will take place in real-world protein design if more properties were desired, we could explore further extensions to the Tanimoto GP used.

As a predominant challenge inherent to *in silico* lead optimization, BOAT accepts oracle predictions as the 'ground truth'. The Pareto front found for computational oracles might only poorly represent the true underlying experimental Pareto front if the predictive power of the oracles is poor. Yet, the discrepancy between *in silico* predictions and experimental measurements, especially for affinity, is a common issue in antibody design, aggravated by data scarcity. Hence, the ultimate experimental performance of BOAT hinges on oracle quality. An exciting yet challenging direction of future research might be to inform BOAT surrogates with experimental data and leverage the learned correlation with oracles to directly optimize experimental properties. Besides data availability, a strong inductive bias will be needed to leverage correlations between experiments and *in silico* predictors successfully.

Acknowledgements

JR acknowledges the support of this work through an internship at AstraZeneca, Cambridge, UK. The authors thank Dino Oglić, Owen Vickery, and Isabelle Sermadiras for valuable discussions, as well as Tom Diethe for constructive feedback on the manuscript.

References

Adams, R. M., Mora, T., Walczak, A. M., and Kinney, J. B. (2016). Measuring the sequence-affinity landscape of antibodies with massively parallel titration curves. *Elife*, 5:e23156.

Ament, S., Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2023). Unexpected improvements to expected improvement for Bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612.

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. *arXiv preprint arXiv:2412.07763*.

Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K., and Colwell, L. (2019). 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. *Advances in Neural Information Processing Systems*, 33:21524–21538.

Binois, M. and Wycoff, N. (2022). A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26.

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

Boder, E. T. and Wittrup, K. D. (1997). Yeast surface display for screening combinatorial polypeptide libraries. *Nature biotechnology*, 15(6):553–557.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. (2022). ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110.

Chungyoun, M., Ruffolo, J., and Gray, J. (2024). FLAb: Benchmarking deep learning methods for antibody fitness prediction. *BioRxiv*, pages 2024–01.

Daulton, S., Balandat, M., and Bakshy, E. (2020). Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864.

Daulton, S., Balandat, M., and Bakshy, E. (2021). Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34:2187–2200.

Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2022). Multi-objective Bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR.

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., 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)*, pages 2147–2154. IEEE.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local Bayesian optimization. *Advances in Neural Information Processing Systems*, 32.

Feliot, P., Bect, J., and Vazquez, E. (2018). User preferences in Bayesian multi-objective optimization: the expected weighted hypervolume improvement criterion. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 533–544. Springer.

Ferruz, N., Schmidt, S., and Höcker, B. (2022). ProtGPT2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348.

Frazier, P. I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Furui, K. and Ohue, M. (2025). ALLM-Ab: Active learning-driven antibody optimization using fine-tuned protein language models. *Journal of Chemical Information and Modeling*, 65(21):11543–11557.

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press.

Gessner, A., Ober, S. W., Vickery, O., Oglić, D., and Uçar, T. (2024). Active learning for affinity prediction of antibodies. *arXiv preprint arXiv:2406.07263*.

Goel, S., Schray, P. M., Zhang, Y., Vincoff, S., Kratochvil, H. T., and Chatterjee, P. (2024). Token-level guided discrete diffusion for membrane protein design. *arXiv preprint arXiv:2410.16735*.

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.

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. *Advances in Neural Information Processing Systems*, 37:140478–140508.

Groth, P. M., Kerrn, M., Olsen, L., Salomon, J., and Boomsma, W. (2024). Kermut: Composite kernel regression for protein variant effects. *Advances in Neural Information Processing Systems*, 37:29514–29565.

Gruver, N., Stanton, S., Frey, N., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. (2023). Protein design with guided discrete diffusion. *Advances in Neural Information Processing Systems*, 36:12489–12517.

He, X.-h., Li, J.-r., Xu, J., Shan, H., Shen, S.-y., Gao, S.-h., and Xu, H. E. (2025). 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.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Katoch, S., Chauhan, S. S., and Kumar, V. (2021). 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., et al. (2022). AntBO: Towards real-world automated antibody design with combinatorial Bayesian optimisation. *arXiv preprint arXiv:2201.12570*.

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. *bioRxiv*, pages 2025–08.

Lee, S., Chu, J., Kim, S., Ko, J., and Kim, H. J. (2023). Advancing Bayesian optimization via learning correlated latent space. *Advances in Neural Information Processing Systems*, 36:48906–48917.

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

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., et al. (2025). DyAb: sequence-based antibody design and property prediction in a low-data regime. *bioRxiv*, pages 2025–01.

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).

Malherbe, C. and Ucar, T. (2024). IgBlend: unifying 3D structures and sequences in antibody language models. *bioRxiv*, pages 2024–10.

Maus, N., Jones, H., Moore, J., Kusner, M. J., Bradshaw, J., and Gardner, J. (2022). Local latent space Bayesian optimization over structured inputs. *Advances in Neural Information Processing Systems*, 35:34505–34518.

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 *International conference on machine learning*, pages 24398–24419. PMLR.

Moss, H., Leslie, D., Beck, D., Gonzalez, J., and Rayson, P. (2020). BOSS: Bayesian optimization over string spaces. *Advances in Neural Information Processing Systems*, 33:15476–15486.

Moss, H. B., Ober, S. W., and Diethe, T. (2025). Return of the latent space COWBOYS: Re-thinking the use of VAEs for Bayesian optimisation of structured spaces. *arXiv preprint arXiv:2507.03910*.

Nana Teukam, Y. G., Zipoli, F., Laino, T., Criscuolo, E., Grisoni, F., and Manica, M. (2025). Integrating genetic algorithms and language models for enhanced enzyme design. *Briefings in bioinformatics*, 26(1):bbae675.

Nigam, A., Friederich, P., Krenn, M., and Aspuru-Guzik, A. (2019). Augmenting genetic algorithms with deep neural networks for exploring the chemical space. *arXiv preprint arXiv:1909.11655*.

Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N., and Madani, A. (2023). ProGen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978.

Notin, P., Hernández-Lobato, J. M., and Gal, Y. (2021). Improving black-box optimization in VAE latent space using decoder uncertainty. *Advances in Neural Information Processing Systems*, 34:802–814.

Oglic, D. and Gärtner, T. (2018). Learning in reproducing kernel Krein spaces. In *International conference on machine learning*, pages 3859–3867. PMLR.

Oglic, D. and Gärtner, T. (2019). Scalable learning in reproducing kernel Krein spaces. In *International Conference on Machine Learning*, pages 4912–4921. PMLR.

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. *Bioinformatics*, 40(11):btae618.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., et al. (2025). Boltz-2: Towards accurate and efficient binding affinity prediction. *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. In *MAbs*, volume 14, page 2020203. Taylor & Francis.

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

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

Ren, M., He, Z., and Zhang, H. (2025). Multi-objective antibody design with constrained preference optimization. In *The Thirteenth International Conference on Learning Representations*.

Ren, Z., Li, J., Ding, F., Zhou, Y., Ma, J., and Peng, J. (2022). Proximal exploration for model-guided protein sequence design. In *International Conference on Machine Learning*, 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., et al. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the national academy of sciences*, 118(15):e2016239118.

Sela-Culang, I., Kunik, V., and Ofran, Y. (2013). The structural basis of antibody-antigen recognition. *Frontiers in immunology*, 4:302.

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.

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. *arXiv preprint arXiv:2010.02141*.

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 *International conference on machine learning*, pages 20459–20478. PMLR.

Tripp, A., Daxberger, E., and Hernández-Lobato, J. M. (2020). Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33:11259–11272.

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., Hutter, F., Zoghi, M., Matheson, D., and De Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387.

Widatalla, T., Rafailov, R., and Hie, B. (2024). Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, pages 2024–05.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Xu, J. L. and Davis, M. M. (2000). Diversity in the CDR3 region of VH is sufficient for most antibody specificities. *Immunity*, 13(1):37–45.

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. *arXiv preprint arXiv:2505.15093*.

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

Zambaldi, V., La, D., Chu, A. E., Patani, H., Danson, A. E., Kwan, T. O., Frerix, T., Schneider, R. G., Saxton, D., Thillaisundaram, A., et al. (2024). De novo design of high-affinity protein binders with AlphaProteo. *arXiv preprint arXiv:2409.08022*.

Zhang, D., Fu, J., Bengio, Y., and Courville, A. (2021). Unifying likelihood-free inference with black-box optimization and beyond. *arXiv preprint arXiv:2110.03372*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **No - time is dominated by the oracle of interest, and complexity of GPs is known.**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Will be made available upon acceptance.**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Not Applicable**
   (b) Complete proofs of all theoretical results. **Not Applicable**
   (c) Clear explanations of any assumptions. **Not Applicable**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Instructions are available. Code will be made available upon acceptance.**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes - throughout Section 4 and in Appendix 1, 7**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes - throughout Section 4**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **No. The main algorithm can be run on a local CPU in seconds, resource requirements depend on oracles used.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**
   (b) The license information of the assets, if applicable. **No. All external packages used are cited and open source.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **No - Code and license will be made available upon acceptance.**
   (d) Information about consent from data providers/curators. **Not Applicable**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

BOAT: Navigating the Sea of In Silico Predictors for Antibody Design via Multi-Objective Bayesian Optimization: Supplementary Material

A FURTHER EXPERIMENTAL DETAILS - CROSS-REACTIVITY

A.1 5 maximum mutations

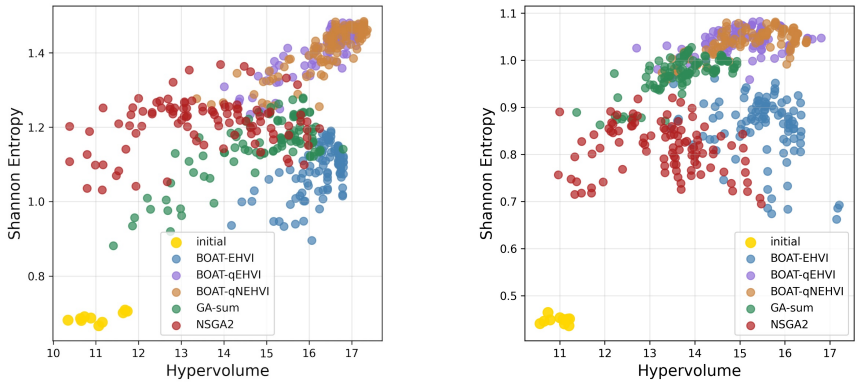
In all experiments, we generate 100 initial sequences with 2 maximum mutations for each of 10 different random seeds per method. All methods were allowed up to 1000 oracle calls to evaluate sequences. Batch acquisition functions (indiciated by a 'q' in the acquisition function; qEHVI and qNEHVI) had 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 4.1 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. 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.

A.2 Structure prediction oracle

We considered a total of 512 oracle calls after the initial 50 evaluations. This translates to 16 rounds of 32 new sequences in the GAs, and 64 iterations of BOAT using a batch size of 8. The GA for optimizing the acquisition function had a budget of 50×50 sequences in every BOAT iteration.

B SHANNON ENTROPY FOR OTHER CDRs



(a) CDR1 (b) CDR2

Figure 8: 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.

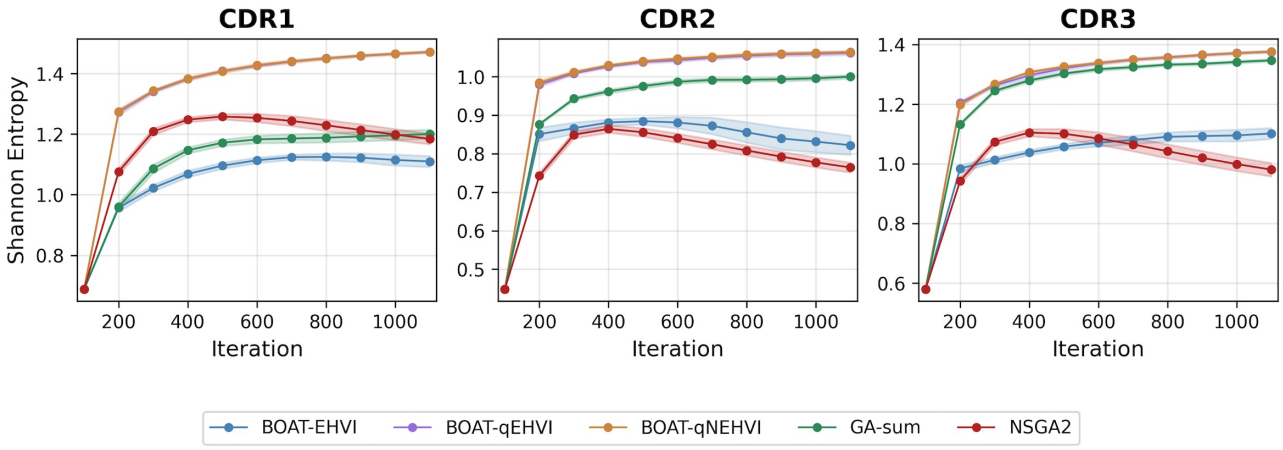
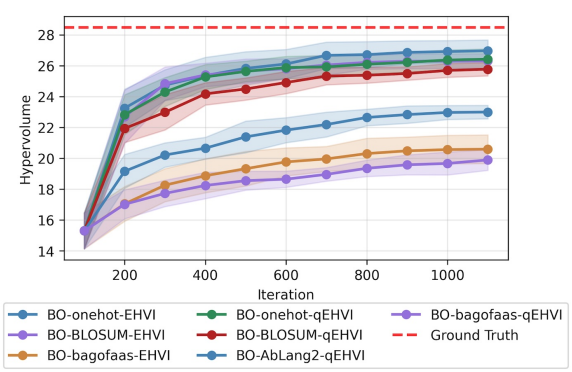


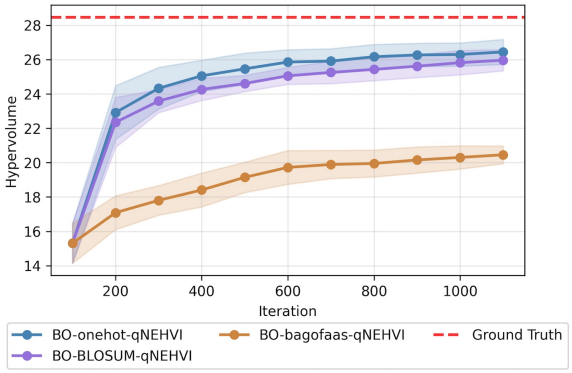
Figure 9: Evolution of Shannon entropy over optimization iterations for CDR1, CDR2, and CDR3 regions. Results averaged over 10 seeds with standard error bands, showing how population diversity changes throughout the optimization process across all three CDR regions. Note BOAT-qEHVI performs almost exactly in line with BOAT-qNEHVI.

C COMPARING ENCODINGS

We compared the performance of different encodings (see Section 4.1) across 10 seeds when optimizing CDR3 for the V_{HH} antibody. We allowed for 5 maximum mutations on CDR3, and optimized for 2 affinity objectives. For each encoding, we tested EHVI, qEHVI and qNEHVI (with batch sizes 4), apart from AbLang-2, where we only tested qEHVI, as the computational cost of querying a language model made it significantly slower than all other encodings; each run of a GA took up to two minutes compared to less than five seconds for other encodings.



(a) Comparisons of encodings for the EHVI and qEHVI (batch size 4) acquisition functions.

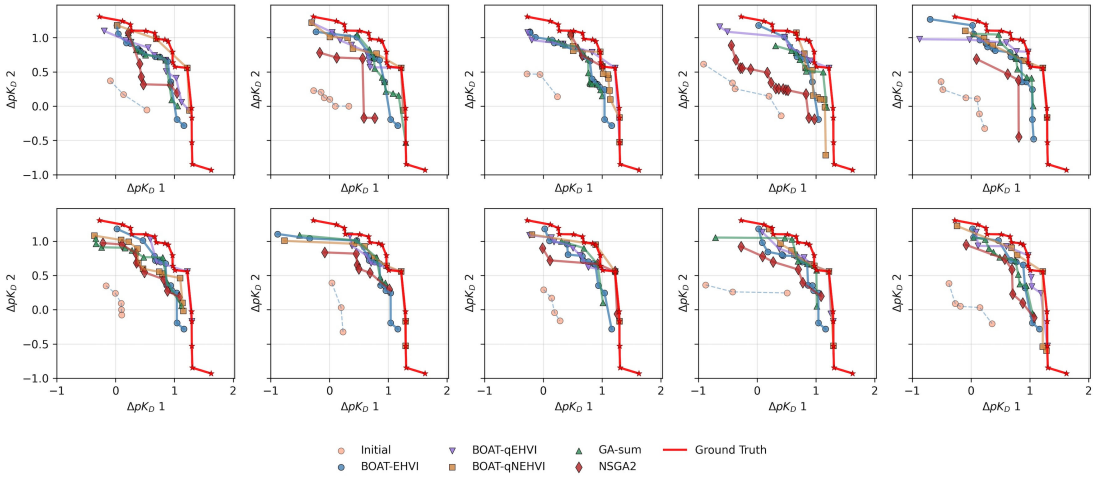


(b) Comparisons of encodings for the qNEHVI acquisition function.

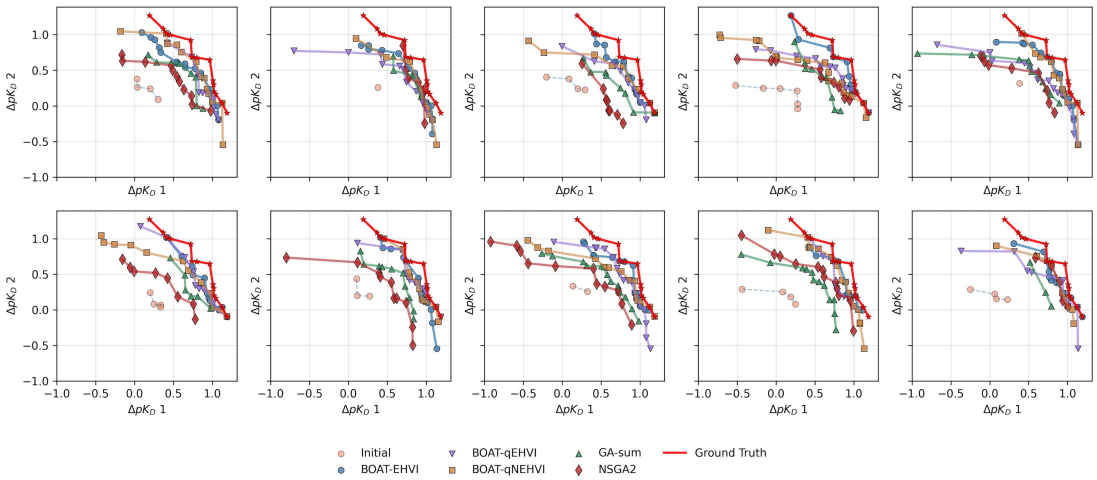
Figure 10: Plots comparing the hypervolume of the Pareto front by iteration of the BO algorithm for different encodings considered in Section 4.1.

We saw in this example that the BLOSUM and one-hot encodings performed similarly. AbLang-2 actually performed worse than both BLOSUM and one-hot over 10 seeds, despite the encoding being more complex. Bag of amino acids was the worst performer of all the encodings. In further experiments, we used either BLOSUM or one-hot encoding, as both were relatively fast and performed equivalently well.

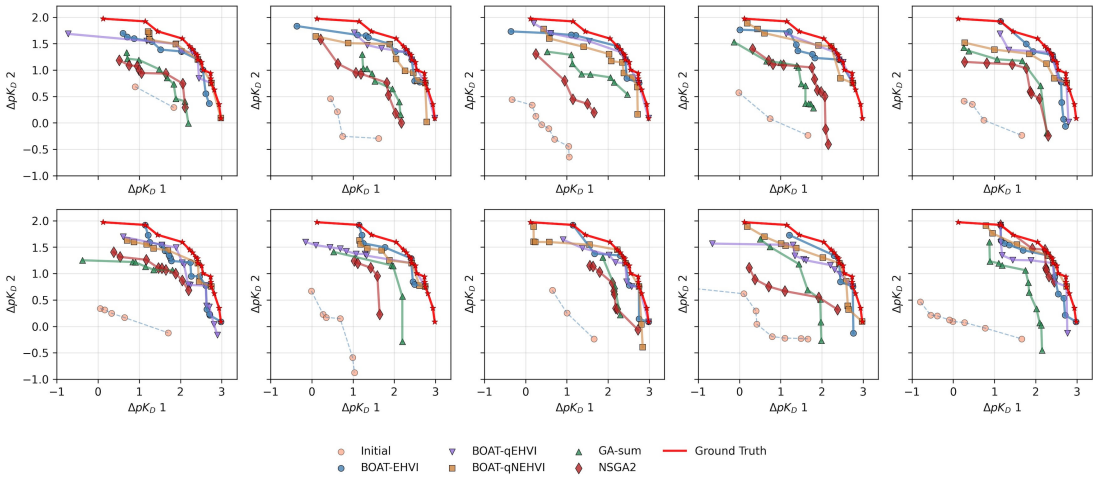
D FULL COMPARISON FOR 5 MUTATIONS FOR ALL SEEDS



(a) CDR1

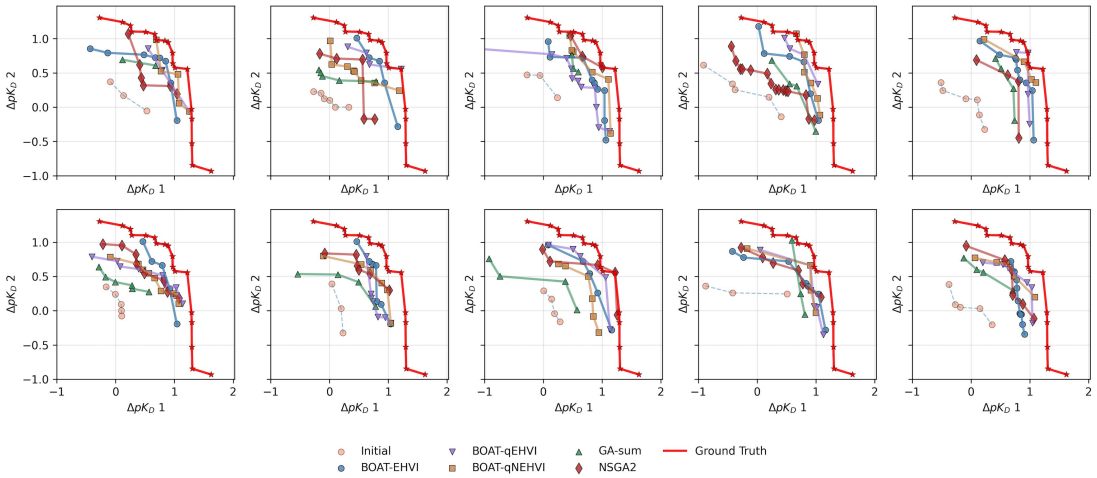


(b) CDR2

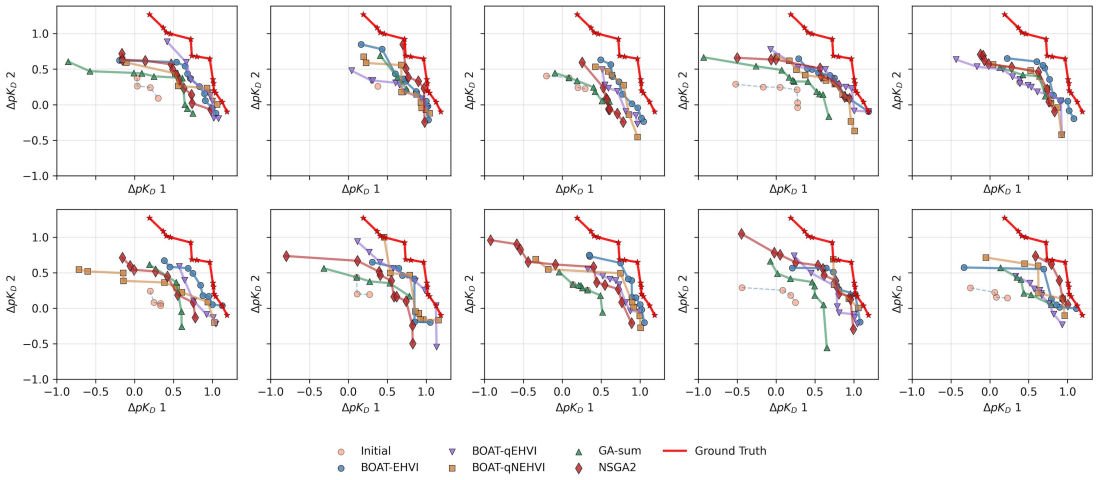


(c) CDR3

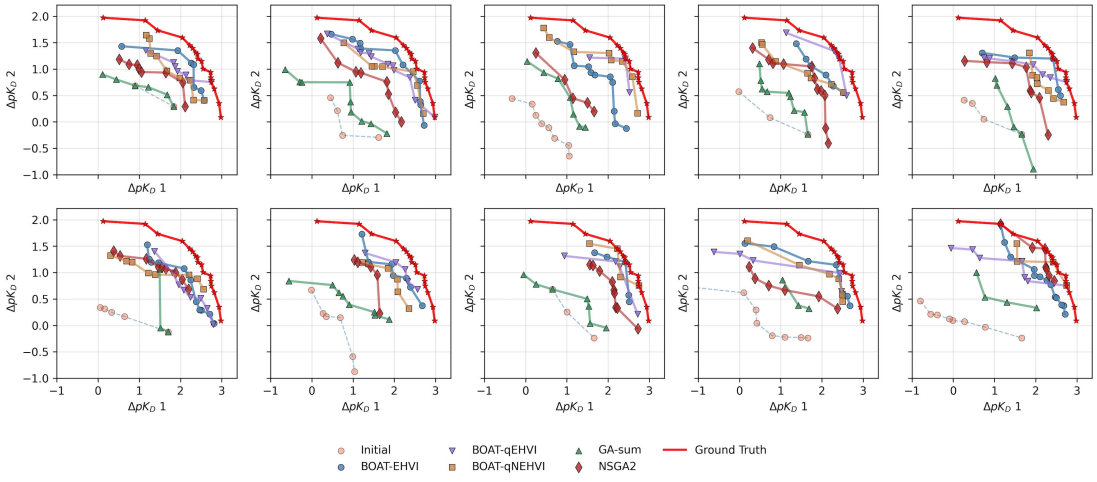
Figure 11: Plot comparing Pareto front by seed across different conditions for each CDR. All seeds are shown. It is clear especially in CDR3 runs that GAs tend to explore smaller Pareto fronts.



(a) CDR1 (200 oracle calls)



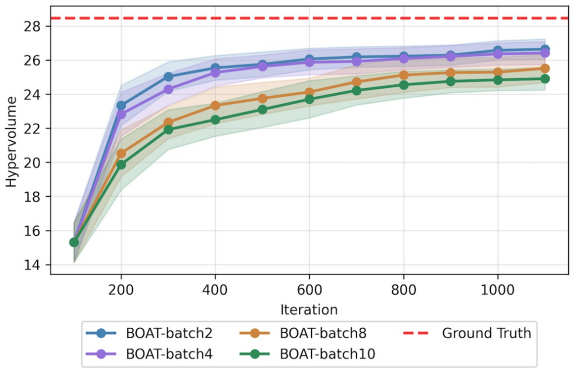
(b) CDR2 (200 oracle calls)



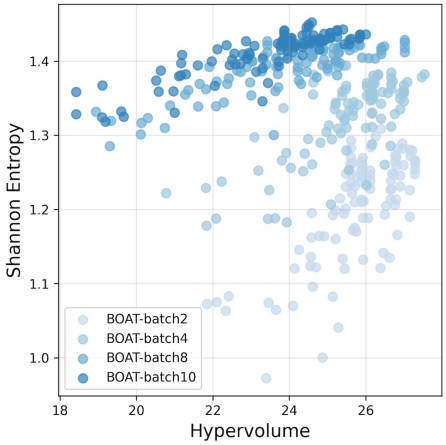
(c) CDR3 (200 oracle calls)

Figure 12: Plot comparing Pareto front after 300 generated sequences (200 oracle calls after 100 initial sequences) by seed across different conditions for each CDR. All seeds are shown. Early in the algorithm (especially for CDR3), BOAT is able to explore much closer to the ground truth Pareto front compared to GAs.

E COMPARING BATCH SIZE



(a) Hypervolume evolution comparison of batch sizes for CDR3, using qEHVI acquisition function.



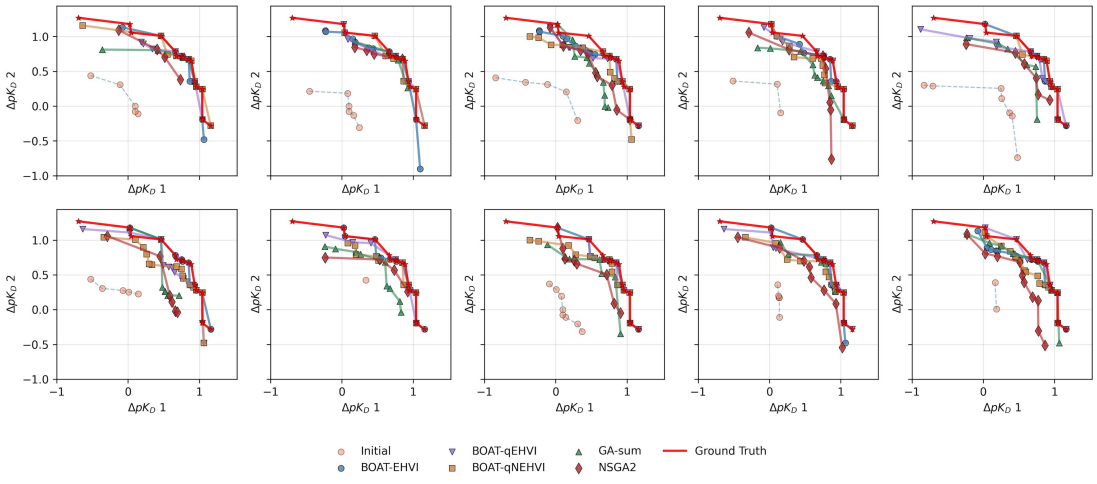
(b) Scatter plots of hypervolume versus Shannon entropy for CDR3 optimization. Initial solutions are not shown.

Figure 13: Plots for experiments when ablating batch size using the qEHVI acquisition function for CDR3 with 2 objectives and 5 maximum mutations, using batch sizes 2, 4, 8, 10. Other experimental details are kept the same as before. For (b), as before, each point represents the diversity and multi-objective performance of a population at a given optimization step, showing results for all seeds and batch sizes, and every 100 iterations.

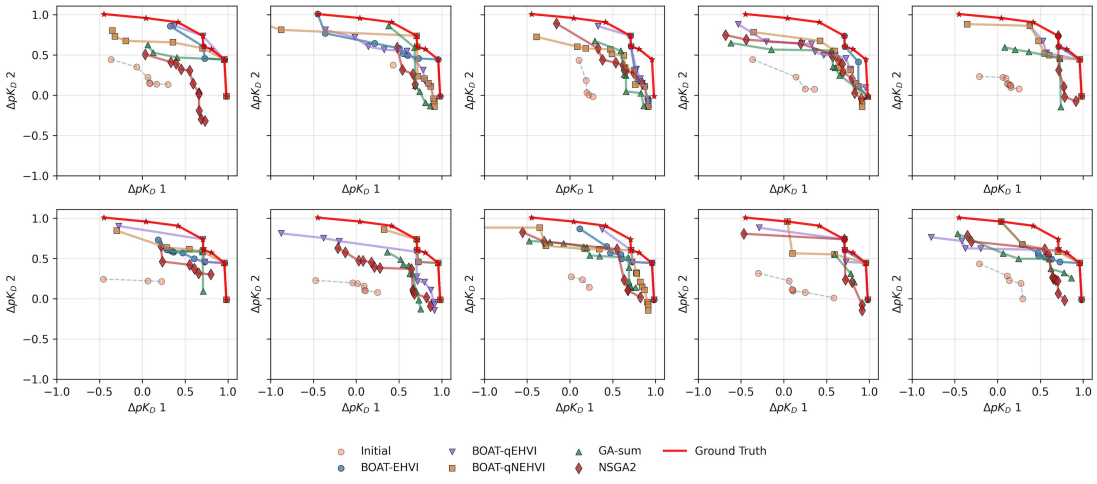
We see in Figure 13a that although we see decreased performance in terms of hypervolume improvement as we increase the batch size for the qEHVI acquisition function - especially towards the beginning of the algorithm - as we perform more iterations, results become more similar, with batch sizes 2 and 4 almost indistinguishable. As expected, we see more diversity as measured by Shannon entropy when using larger batch sizes, as seen in Figure 13b.

F 4 MAXIMUM MUTATIONS

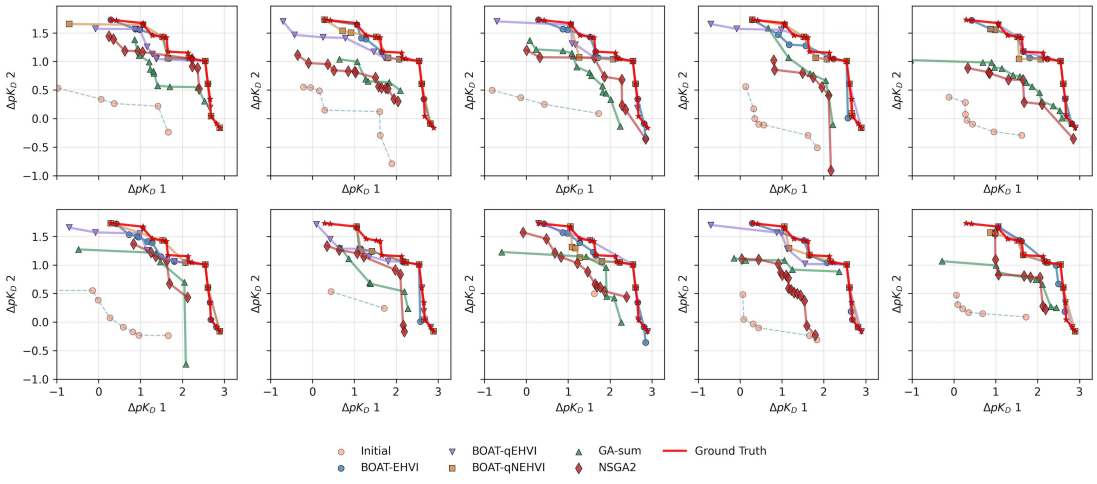
We ran a similar experiment to that in the main paper for 5 maximum mutations, but instead with 4 maximum mutations, which had a much smaller search space - CDR3 had 3975741 ground truth sequences, for example. We visualize the Pareto fronts found versus the ground truth in Figure 14.



(a) CDR1



(b) CDR2



(c) CDR3

Figure 14: Plot comparing Pareto front by seed across different conditions for each CDR. All seeds are shown. Similarly to the 5 mutations example, GAs visually do not explore as much of the ground truth Pareto front compared to BOAT.

G COMPARISON WITH LaMBO-2

G.1 Details on the experimental setup

Training of the discriminative head for LaMBO-2 was run for 100 epochs. For longer training, we observed the validation error increase. 561 sequences were held out for testing. We optimized 16 seed antibodies, which were all set to be the wild-type sequence, and ran LaMBO for 8 steps (with 4 guidance updates per step), leading to 256 total sequences being generated and evaluated. All other parameters followed default settings from the *cortex* GitHub implementation of LaMBO-2. For numerical stability and to phrase the optimization problem as an unconstrained maximization problem, we take the negative logarithm of the measured k_D values and log-transform the expression data.

To provide BOAT with a fair comparison, we used LaMBO-2's trained discriminative head f^* as the black-box objectives for BOAT (predicting affinity and expression). We initialized with 16 sequences containing up to 2 mutations from the wild-type, scored these with f^*, then ran BOAT for 256 iterations in the sequential setting using EHVI, and for 64 iterations with a batch size of 4, i.e., also 256 oracle calls, for qEHVI. Given the modest training dataset size and corresponding uncertainty in the discriminative head's predictions, we also ran qNEHVI to explicitly model the noise in objective function evaluations with the same batch size as qEHVI. We ran both models with 5 different seeds. In order to maintain comparability across seeds, we trained the discriminative head once and then varied seeds across LaMBO and BOAT optimization runs. To address the naturalness of the sequences which is in-built for LaMBO-2, we explore two settings for BOAT, 1) a 3-objective setting in which we include ESM-2 as a third objective, 2) not including any naturalness constraint, i.e., only considering the given affinity and expression oracles.

G.2 Additional results

G.2.1 With naturalness constraint on BOAT

The inclusion of a PLM likelihood can be seen as a "regularizer" on sequences, as sequences that are more likely to be found in nature will receive higher scores. We ran the optimization for BOAT-EHVI, BOAT-qEHVI, and BOAT-qNEHVI. However, in the 3-objective setting, the qNEHVI is too slow to be recommended for practical applications. For the 5 seeds tested, we plot the final Pareto front for the BOAT variants, all LaMBO designs, the original wild-type point (with its predicted affinity and expression), as well as the designs found by BOAT-EHVI in Figure 15. These are the designs corresponding to the final state of the hypervolume progression plotted in Figure 7.

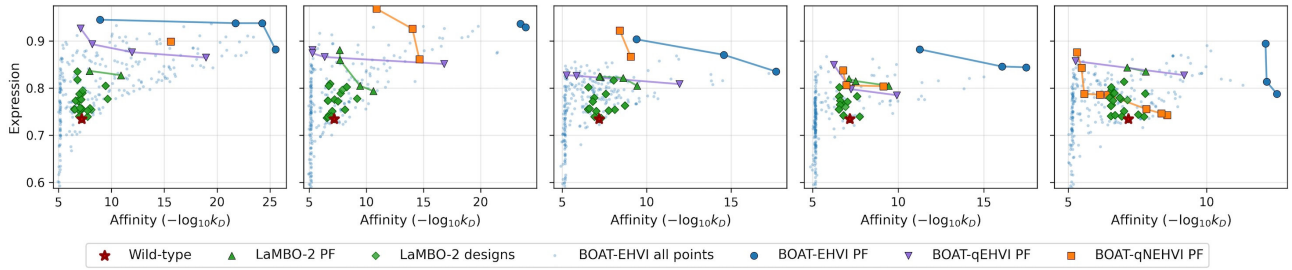


Figure 15: Multi-seed comparison of BOAT and LaMBO showing final Pareto fronts for BOAT-EHVI, BOAT-qEHVI, BOAT-qNEHVI, and all LaMBO designs across 5 optimization runs. While LaMBO got to optimize the 2 objectives for affinity and expression, the BOAT variants additionally optimized for naturalness (ESM-2 likelihood). As the hypervolumes are fully comparable across seeds, we sorted them by descending terminal hypervolume for each method.

Figure 15 demonstrates the capability of all BOAT methods to push the Pareto front, while LaMBO-2 designs exhibit a less explorative behaviour and produce sequences that are more similar to the parent. BOAT explores a lot more aggressively than LaMBO-2 by mutating further away from the parental at a faster rate. The plots also illustrate issues of the discriminative head, which predicts unrealistically large values for affinity for some sequences. It can be further seen that many of the BOAT-generated designs seem to be close to an affinity cut-off

at 5. This is due to the dataset containing many sequences with $-\log_{10} k_D = 5$, which is likely the value assigned to all sequences without measurable binding. Remarkably, BOAT-EHVI finds the sequences with unrealistically low predicted k_D values across seeds, while the batch versions do not. We postulate that the combinatorially larger amount of candidate batches in the batch versions makes it less likely to select one a batch containing one of these extreme but rare sequences than in the sequential case.

G.2.2 Without naturalness constraint on BOAT

When omitting naturalness constraints on BOAT, it becomes obvious in Figure 16 that BOAT variants aggressively push the hypervolume by evaluating sequences with unrealistic affinity values caused by overfitting in the discriminative head. Yet, this highlights that BOAT in principle has the capability of finding such interesting sequences in more trustworthy oracles. LaMBO-2 is instead more conservative in optimizing leads, which prevents it from falling into the pitfall of proposing out-of-distribution sequences, but also keeps it from exploring more diverse sequences.

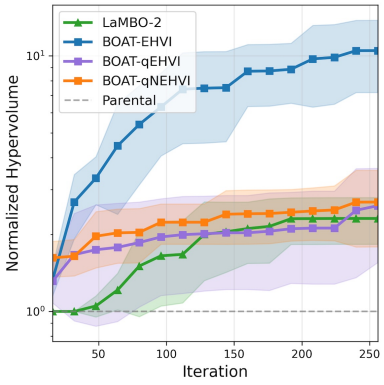


Figure 16: Hypervolume progression for LaMBO-2 and BOAT variants when omitting naturalness constraints on BOAT.

Figure 17 shows the Pareto fronts corresponding to the final hypervolumes in Figure 16. Even more than in the naturalness-constrained setting discussed in Section G.2.1, we observe the sequential version of BOAT to discover sequences where the discriminative head fails.

The obvious quality issues of LaMBO-2's discriminative head underlines the benefit of the modular design of BOAT, which permits including tailored oracles for particular properties. While LaMBO-2 can be seen as an elegant lab-in-the-loop approach which requires little human intervention, the limitation to a fixed architecture for the discriminative head can also be seen as a defect that impedes leveraging auxiliary information that is not present in the data used for training. Not only does LaMBO-2 not overcome the need for human oversight and we observed, it requires profound technical understanding to perform low-level adaptations on the model to given task. With the possibility to interface externally built and validated oracles, we claim that BOAT does not require the same depth of technical understanding to run.

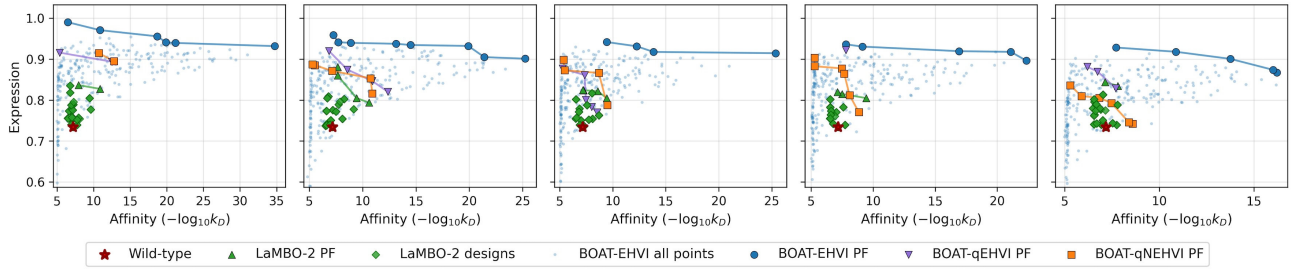


Figure 17: Multi-seed comparison of BOAT and LaMBO showing final Pareto fronts for BOAT-EHVI, BOAT-qEHVI, BOAT-qNEHVI, and all LaMBO designs across 5 optimization runs, where both LaMBO and BOAT got to optimize the two objectives of affinity and expression predicted by LaMBO's discriminative head. As the hypervolumes are fully comparable across seeds, we sorted them by descending terminal hypervolume for each method.