WHY RISK MATTERS FOR PROTEIN BINDER DESIGN

Tudor-Stefan Cotet[1,2] **& Igor Krawczuk**[1]
[1]Adaptyv, Lausanne, Switzerland
[2]Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland

ABSTRACT

Bayesian optimization (BO) has recently become more prevalent in protein engineering applications and hence has become a fruitful target of benchmarks. However, current BO comparisons often overlook real-world considerations like risk and cost constraints. In this work, we compare 72 model combinations of encodings, surrogate models, and acquisition functions on 11 protein binder fitness landscapes, specifically from this perspective. Drawing from the portfolio optimization literature, we adopt metrics to quantify the cold-start performance relative to a random baseline, to assess the risk of an optimization campaign, and to calculate the overall budget required to reach a fitness threshold. Our results suggest the existence of Pareto-optimal models on the risk-performance axis, the shift of this preference depending on the landscape explored, and the robust correlation between landscape properties such as epistasis with the average and worst-case model performance. They also highlight that rigorous model selection requires substantial computational and statistical efforts.

1 INTRODUCTION

Risk in protein optimization: In portfolio optimization, the expected value of a portfolio is often an incomplete measure of its desirability: we may not only care about the average of the results, but also their risk (Gunjan & Bhattacharyya, 2023). We can imagine 2 portfolios that both return the same expected value: one of them returns 10% of average in its worst outcomes and the other returns 90%. It is pretty clear now which one is more desirable.

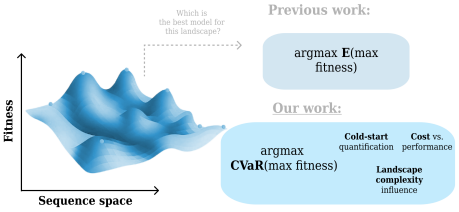

Figure 1: Our additions for protein optimization benchmarking: we consider risks, costs, and performances relative to a random baseline.

This trade-off also occurs in Bayesian optimization (Cakmak et al., 2020; Makarova et al., 2021). Initialization of parameters or input space, inherent experimental noise, and inadequate exploration can make results highly stochastic (Tripp & Hernández-Lobato, 2024; Wang et al., 2022). In protein optimization campaigns, failures (not reaching a target fitness) mean more resources such as time and money will be spent. Thus, when benchmarking models to be used in real campaigns, we should account for and mitigate the risk of failure induced by this stochasticity as much as possible.

Previous work: Recent work (Yang et al., 2025; Li et al., 2024; Jiang et al., 2024) in active learning and Bayesian optimization for proteins has focused on evaluating the performance of the model on landscapes from the ProteinGym (Notin et al., 2023) benchmark (although biological test functions are becoming more prevalent, as introduced by Chen et al. (2024) and Stanton et al. (2024)). These works then selected the best-performing one to use in an experimental campaign according to the final fitness reached (Yang et al., 2025) or fold-change from the starting library (Jiang et al., 2024).

To our knowledge, no benchmark for protein optimization explicitly calculated risks, nor addressed budget considerations (as $ amount) or cold-start performance. This motivates us to study two main questions:

1. **How important is risk analysis when benchmarking and selecting Bayesian optimization algorithms for binders?** For this, we investigate the cost savings when ranking models by the CVaR (conditional value at risk) and average performance, and ensure the statistical significance of our results by performing a bootstrap analysis. First, we simulate optimization campaigns for 72 model combinations on 11 binding datasets. We show that, currently, there is no added benefit of a risk-aware ranking, as this is overshadowed by the inherent stochasticity of protein optimization.

2. **Does the fitness landscape influence the average model performance, risks, and costs?** To answer this, we assess the correlation between risk and landscape properties. With the same robust statistical analysis, we show that both the average and CVaR final fitness and costs are greatly influenced by epistasis.

2 METHODS

2.1 PROBLEM DEFINITION

Inputs: Let $x \in X$ denote a protein sequence of length L, where X is the space of all possible sequences composed of amino acids from an alphabet A (typically, $|A| = 20$). We represent protein sequences using embeddings from a protein language model or with one-hot encodings. Let $e = \phi(x) \in \mathbb{R}^D$ denote the encoding of the sequence x, where $\phi : X \to \mathbb{R}^D$ is the embedding function and D is the embedding dimensionality.

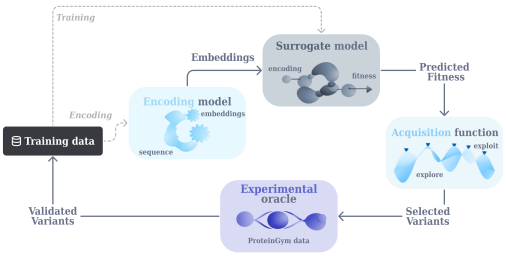

Figure 2: Overview of the standard protein BO loop we are benchmarking.

The black-box function: In an optimization campaign, our goal is to find the protein sequence x^* that maximizes the unknown fitness function f. Since f is expensive to evaluate (i.e., requires performing an assay), Bayesian optimization is commonly used to iteratively propose sequences to evaluate, balancing exploration and exploitation (Frazier, 2018). In the context of protein engineering, f represents binding screening experiments or binding affinity measurements, establishing the ground truth fitness values. These can be direct protein-protein binding affinity measurements or more complex proxies, depending on the screening platform.

Surrogates to approximate experimental results: We use a probabilistic surrogate model to approximate f based on observed data $D_n = \{(x_i, e_i, y_i)\}_i^n$, where $y_i = f(x_i) + \epsilon_i$ represents the fitness (i.e., binding affinity) of the oracle and $\epsilon_i \sim N(0, \sigma^2)$ represents the observation noise. Thus, we train a surrogate $\hat{f}$ to predict the fitness value $\hat{y} = \hat{f}(e_i)$.

Acquisition functions: At each iteration, we select the next batch of sequences to evaluate that maximize (Wilson et al., 2018) an acquisition function $\alpha(e;\ D_n)$:

$$e_{new\ batch} = \underset{e_{candidates} \in \mathbb{R}^D}{\arg\max}\ \alpha(e_{candidates}; D_n)$$

Generator and oracle: Our pool of candidates $e_{candidates}$ consists of all available sequences in the ground-truth dataset that have not been acquired until the current iteration. Other methods generated all possible mutants, then queried the true label using an L1 distance to the closest in the ground-truth dataset (Yang et al., 2025).

2.2 EXPERIMENTAL SETUP AND METRICS

Datasets and BO components: We simulate Bayesian optimization campaigns on 11 binding deep-mutational scanning (DMS) datasets from the ProteinGym repository (Notin et al., 2023), and calculated landscape complexity properties (epistasis, skewness, kurtosis, etc.) for these, described in Appendix A.2. We used 6 different uncertainty-aware surrogates: a deep neural network ensemble (ensemble NN), deep neural network with Monte Carlo dropout (dropout NN, Gal & Ghahramani (2015)), a random forest (RF) model (Breiman, 2001; Louppe, 2014), a Gaussian process (GP) model with assumed homoskedastic noise, a deep kernel Gaussian process (Wilson et al., 2015), and a Bayesian neural network (BNN, Jospin et al. (2020)). Architecture and hyperparameter optimization details are summarized in Appendix A.1. For the acquisition functions, we implemented greedy top-K, Thompson sampling (TS), expected improvement (EI), and upper-confidence bound (UCB, Srinivas et al. (2009)). We tested one-hot encodings, ESM2-650M, and ESM-3B embeddings mean-pooled Lin et al. (2023).

Fitness metrics: Our main performance metric is the normalized maximum fitness reached at the end of a campaign, commonly used in previous work (Li et al., 2024; Zhang et al., 2024), and its conditional value at risk (CVaR) for the worst 10% of cases. We refer you to Mitra & Ji (2009) and Artzner et al. (1999) for an overview of risk measures in financial mathematics. Additional details about our implementation can be found in Appendix A.3.

Quantifying cold-starts: We have included an additional performance metric accounting for cold-starts, a case in which the model performing similarly to a random baseline in the first steps due to an unadapted surrogate and a small (to none) set of initial training points (Poloczek et al., 2016). For this, we calculated the difference between the maximum fitness reached by a model versus the random baseline (ΔG), further computing its area-under-the-curve ($\Delta G\ AUC$). This is a measure of how effectively the model can adapt to and exploit the landscape structure.

Quantifying costs: Costs were assessed considering an acquisition budget of 384 sequences and a 96 starting pool. Thus, assuming a \$150 price point for testing a single variant [1], our maximum acquisition budget will be \$57,600 (\$72,000 with the seed library). We have assessed the cost to reach the 99th percentile of fitness for each landscape. This threshold can depend on the priorities of each campaign and we might be more interested in a fold improvement compared to the starting pool for real-life scenarios.

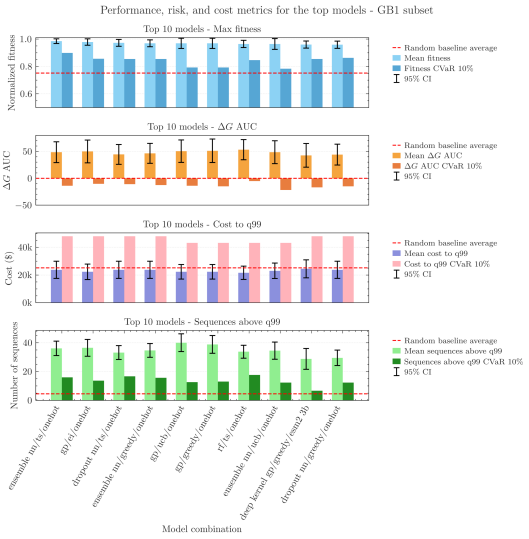

Figure 3: Metrics for the top 10 models ranked by average final fitness for GB1: final fitness reached, $\Delta G\ AUC$, cost to 99th percentile of fitness, number of sequences above the 99th percentile threshold acquired.

3 RESULTS

Returning to our two questions, we first compare CVaR and average-based model rankings (1), showcase that these agree more on the GB1 landscape, and then determine which landscape property is the best predictor of optimization risk (2).

[1] Based on the cheapest result found when Googling "cheapest binding affinity characterization service wetlab" as of January 2025

There is a clear risk-performance model preference on the GB1 landscape: The GB1 landscape (Olson et al., 2014; Wu et al., 2016) has been a workhorse for benchmarking protein optimization algorithms (Li et al., 2024; Yang et al., 2025; Greenman et al., 2025). It explores the interaction effects at 4 sites in the IgG-binding domain of protein G (GB1), yielding almost 160,000 total variants. To reduce the computational complexity, we fixed the residue V54 to the wild-type, resulting in 7600 total sequences and 6080 without the subset used for the surrogate hyperparameter search, as detailed in Appendix A.1 and A.2.

When ranking models by average final fitness (Supplementary Figure A.8), our benchmark recovers the same optimal model identified by the ALDE framework (Yang et al., 2025): DNN ensemble with Thompson sampling and one-hot encodings. In Figure 3, we have summarized several outcome metrics and their associated CVaR values.

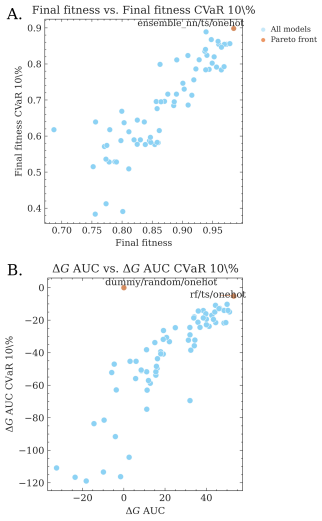

Figure 4: Pareto frontier when considering the performance-risk axes for the final fitness reached and for the $\Delta G\ AUC$ metric.

This multi-metric analysis reveals a fundamental tension: while GB1's structure allows clear identification of better models under any single metric, different optimization goals can lead to completely different model preferences. For example, the GP/UCB/one-hot model achieves a competitive final fitness, but has a lower CVaR (Figure 3). It still yields the highest number of sequences above the 99th percentile of fitness on average. This suggests that even for well-studied landscapes, model selection should account for the end goals of the optimization campaign. In a typical industrial setting, a risk-aware selection could be more suitable, which we will assess next.

The best model is Pareto-dominant when looking at the risk-performance axis (Figure 4). The optimal model differs for $\Delta G\ AUC$ (Figure 4). The random forest model might be preferable if we want both to optimize and to have a model adequately learn the structure of a landscape. Several other models become Pareto-optimal when costs are taken into account (Supplementary Figure A.19. These leverage Bayesian NN surrogates and can easily achieve the fitness threshold set, yet they are often not optimizing past it.

Dataset	Kendall τ
HLA-A	0.549^{***}
CD19	0.252^{**}
CCR5	0.212^{**}
ACE2	0.681^{***}
CytochromeP4502C9	0.568^{***}
Dlg4 PSD95 PDZ3	0.606^{***}
KRAS	0.544^{***}
GB1 subset	0.667^{***}
YAP1	0.769^{***}
SpikeRBD	0.375^{***}
Gcn4	0.149
All datasets	0.473^{***}

Table 1: Kendall τ correlation between CVaR and average-based model ranks for each landscape. *** denotes a p-value < 0.001, ** for < 0.01, and * for < 0.05.

Variability in landscapes and optimization limits risk-aware model selection: Next, we looked at the rank correlations for the 72 models in each landscape (Table 1). The GB1 subset displays high agreement, while others (e.g., CCR5) show that model preferences vary drastically when accounting for risk in our rankings. We hypothesized that a risk-aware ranking might be more cost-efficient, ideally reducing worst-case costs, and that more complex landscapes would benefit from a risk-aware ranking.

To compare risk-aware and mean-based model selection, we have performed a bootstrap analysis on the 20 runs to better simulate subsequent optimization campaigns, summarized in Appendix A.4. While naive bootstrapping suggested large potential cost savings (up to $16,000), a more rigorous testing revealed no statistically significant benefits when accounting for risk into the rankings (Supplementary Figure A.6). This suggests we cannot judge based on the simulation whether the risk-aware model choice would actually save money on average or in the worst case. It will require additional seeds and/or k-fold validation, which is a current limitation of our work.

Epistasis is the main driver of increased campaign risks and costs: Next, we have addressed the second question 2 (Figure 5). Bootstrap analysis details are summarized in Appendix A.4. The final fitness has a strong negative correlation with non-magnitude epistasis (-0.52 average, -0.55 CVaR, statistically significant), while showing moderate negative correlations with kurtosis (-0.39/-0.43), the number of KDE peaks (-0.38/-0.46), and skewness (-0.38/-0.34). This indicates that landscape complexity is directly detrimental to the final performance of all models. The worst 10% of cases get marginally worse with increased complexity compared to the average. The percentage of active variants shows moderate positive correlations (0.37/0.33), suggesting that an abundance of active variants can stabilize performance.

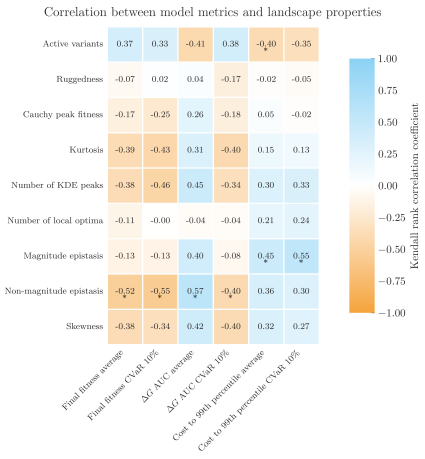

Figure 5: Kendall τ correlation coefficients between the average model metrics and landscape properties. *** denotes a p-value < 0.001, ** for < 0.01, and * for < 0.05.

Several landscape complexity metrics correlate positively with the average $\Delta G\ AUC$. Most notably, non-magnitude epistasis (0.57), the number of KDE peaks (0.45), magnitude epistasis (0.40), and skewness (0.42) show substantial positive correlations. However, these metrics generally correlate negatively with the $\Delta G\ AUC$ CVaR 10%, as seen with non-magnitude epistasis (-0.40) and kurtosis (-0.40), suggesting that while distinct landscape topologies are learnable and exploitable on average by models, they can be detrimental when models do not adapt fast enough. Magnitude epistasis shows the strongest positive correlation to the cost (0.45, statistically significant, but a larger confidence interval, Supplementary Figure A.7) and displays an even stronger correlation in the worst 10% of cases (0.55). Active variants show consistent negative correlations with costs (-0.40/-0.35), indicating that a higher likelihood of reaching such variants reduces optimization costs.

Of all correlations, epistasis is the most significant predictor of risk, performance against random, and costs in an optimization campaign, and should be taken into account before starting an optimization campaign.

4 Conclusion

In this study, we established the importance of risk quantification and statistically robust benchmarking for protein binder optimization and showed that **risk-aware BO benchmarking** does indeed give important additional information (1). We observed reordering effects and breakdown of clear Pareto-dominance when considering risk, costs and performance compared to a random baseline. While rigorous statistical testing did not support the claim of cost savings when changing model selection accordingly, this also remains a tantalizing possibility.

We also showed that there is indeed an **influence of landscape properties**(2) and complex landscapes which present an interesting dichotomy: when learned (on average) they yield better-than-random performance, but they correlate with more catastrophic worst-case performance regressions (when learning fails). We find that epistasis is the best predictor of increased risks and costs for the datasets considered.

Limitations: We only evaluated single-objective risk due to the nuances involved in considering multiple objectives. Due to computational constraints, we had to subset some of our datasets, which might change the results. Many more seeds are required to make reliable statistical claims about the impact of risk on model choice. Our model ranking and evaluation approach would benefit from proper k-fold validation using more compute or data for robustness. Finally, we are working with finite datasets where we can only select variants rather than generate new ones. This limits our ability to understand optimization breakdowns in open-ended campaigns. Subsequent work could implement pre-trained fitness oracles or biological test functions to study these.

REFERENCES

Philippe Artzner, Freddy Delbaen, Jean Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9:203–228, 7 1999. ISSN 1467-9965. doi: 10.1111/1467-9965.00068. URL https://onlinelibrary.wiley.com/doi/full/10.1111/1467-9965.00068https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9965.00068https://onlinelibrary.wiley.com/doi/10.1111/1467-9965.00068.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001. ISSN 08856125. doi: 10.1023/A:1010933404324/METRICS. URL https://link.springer.com/article/10.1023/A:1010933404324.

Sait Cakmak, Raul Astudillo, Peter Frazier, and Enlu Zhou. Bayesian optimization of risk measures. *Advances in Neural Information Processing Systems*, 2020-December, 7 2020. ISSN 10495258. URL https://arxiv.org/abs/2007.05554v3.

Angelica Chen, Samuel D Stanton, Robert G Alberstein, Andrew M Watkins, Richard Bonneau, Vladimir Gligorijevi, Kyunghyun Cho, and Nathan C Frey. Llms are highly-constrained biophysical sequence optimizers. 10 2024. URL https://arxiv.org/abs/2410.22296v3.

Christian Dallago, Jody Mou, Kadina E Johnston BBE, Bruce J Wittmann BBE, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pp. 2021.11.09.467890, 1 2022. doi: 10.1101/2021.11.09.467890. URL https://www.biorxiv.org/content/10.1101/2021.11.09.467890v2https://www.biorxiv.org/content/10.1101/2021.11.09.467890v2.abstract.

Aaron Defazio, Xingyu Alice Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. 5 2024. URL https://arxiv.org/abs/2405.15682v4.

Peter I. Frazier. A tutorial on bayesian optimization. 7 2018. URL https://arxiv.org/abs/1807.02811v1.

Trevor S. Frisby and Christopher James Langmead. Bayesian optimization with evolutionary and structure-based regularization for directed protein evolution. *Algorithms for Molecular Biology*, 16:1–15, 12 2021. ISSN 17487188. doi: 10.1186/S13015-021-00195-4/FIGURES/7. URL https://almob.biomedcentral.com/articles/10.1186/s13015-021-00195-4.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *33rd International Conference on Machine Learning, ICML 2016*, 3:1651–1660, 6 2015. URL https://arxiv.org/abs/1506.02142v6.

Kevin P Greenman, Ava P Amini, and Kevin K Yang. Benchmarking uncertainty quantification for protein engineering. *PLOS Computational Biology*, 21:e1012639, 1 2025. ISSN 1553-7358. doi: 10.1371/JOURNAL.PCBI.1012639. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1012639.

Abhishek Gunjan and Siddhartha Bhattacharyya. A brief review of portfolio optimization techniques. *Artificial Intelligence Review*, 56:3847–3886, 5 2023. ISSN 15737462. doi: 10.1007/S10462-022-10273-7/METRICS. URL https://link.springer.com/article/10.1007/s10462-022-10273-7.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R. Sgrizzi, Lukas Villiger, Alisan Kayabolen, Byungji Kim, Josephine K. Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, Jonathan S. Gootenberg, and Omar O. Abudayyeh. Rapid protein evolution by few-shot learning with a protein language model. *bioRxiv*, pp. 2024.07.17.604015, 7 2024. doi: 10.1101/2024.07.17.604015. URL https://www.biorxiv.org/content/10.1101/2024.07.17.604015v1https://www.biorxiv.org/content/10.1101/2024.07.17.604015v1.abstract.

Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on bayesian neural networks – a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17:29–48, 7 2020. ISSN 15566048. doi: 10.1109/mci.2022.3155327. URL https://arxiv.org/abs/2007.06823v3.

Francesca-Zhoufan Li, Jason Yang, Kadina E. Johnston, Emre Gürsoy, Yisong Yue, and Frances H. Arnold. Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes. *bioRxiv*, pp. 2024.10.24.619774, 10 2024. doi: 10.1101/2024.10.24.619774. URL https://www.biorxiv.org/content/10.1101/2024.10.24.619774v1https://www.biorxiv.org/content/10.1101/2024.10.24.619774v1.abstract.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379:1123–1130, 3 2023. ISSN 10959203. doi: 10.1126/SCIENCE.ADE2574/SUPPL_FILE/SCIENCE.ADE2574_SM.PDF. URL https://www.science.org/doi/10.1126/science.ade2574.

Gilles Louppe. Understanding random forests: From theory to practice. 7 2014. URL https://arxiv.org/abs/1407.7502v3.

Anastasiia Makarova, Ilnura Usmanova, Ilija Bogunovic, and Andreas Krause. Risk-averse heteroscedastic bayesian optimization. *Advances in Neural Information Processing Systems*, 21:17235–17245, 11 2021. ISSN 10495258. URL https://arxiv.org/abs/2111.03637v1.

Sovan Mitra and Tong Ji. Risk measures in quantitative finance. *International Journal of Business Continuity and Risk Management*, 1:125, 4 2009. ISSN 1758-2164. doi: 10.1504/ijbcrm.2010.033634. URL https://arxiv.org/abs/0904.0870v1.

Pascal Notin, Aaron W. Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Hansen Spinner, Nathan Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, and Debora S. Marks. Proteingym: Large-scale benchmarks for protein design and fitness prediction. *bioRxiv*, pp. 2023.12.07.570727, 12 2023. doi: 10.1101/2023.12.07.570727. URL https://www.biorxiv.org/content/10.1101/2023.12.07.570727v1https://www.biorxiv.org/content/10.1101/2023.12.07.570727v1.abstract.

C. Anders Olson, Nicholas C. Wu, and Ren Sun. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current biology : CB*, 24:2643, 2014. ISSN 09609822. doi: 10.1016/J.CUB.2014.09.072. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4254498/.

Matthias Poloczek, Jialei Wang, and Peter I. Frazier. Warm starting bayesian optimization. *Proceedings - Winter Simulation Conference*, 0:770–781, 8 2016. ISSN 08917736. doi: 10.1109/WSC.2016.7822140. URL https://arxiv.org/abs/1608.03585v1.

Jacob T. Rapp, Bennett J. Bremer, and Philip A. Romero. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nature Chemical Engineering 2024 1:1*, 1:97–107, 1 2024. ISSN 2948-1198. doi: 10.1038/s44286-023-00002-4. URL https://www.nature.com/articles/s44286-023-00002-4.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *IEEE Transactions on Information Theory*, 58:3250–3265, 12 2009. doi: 10.1109/TIT.2011.2182033. URL http://arxiv.org/abs/0912.3995http://dx.doi.org/10.1109/TIT.2011.2182033.

Samuel Stanton, Robert Alberstein, Nathan Frey, Andrew Watkins, and Kyunghyun Cho. Closed-form test functions for biophysical sequence optimization algorithms. 6 2024. URL https://arxiv.org/abs/2407.00236v1.

Austin Tripp and José Miguel Hernández-Lobato. Diagnosing and fixing common problems in bayesian optimization for molecule design. 6 2024. URL https://arxiv.org/abs/2406.07709v2.

Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. Recent advances in bayesian optimization. *ACM Computing Surveys*, 55:25, 6 2022. ISSN 15577341. doi: 10.1145/3582078. URL https://arxiv.org/abs/2206.03301v2.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, pp. 370–378, 11 2015. URL https://arxiv.org/abs/1511.02222v1.

James T. Wilson, Frank Hutter, and Marc Peter Deisenroth. Maximizing acquisition functions for bayesian optimization. *Advances in Neural Information Processing Systems*, 2018-December: 9884–9895, 5 2018. ISSN 10495258. URL https://arxiv.org/abs/1805.10196v2.

Nicholas C. Wu, Lei Dai, C. Anders Olson, James O. Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5, 7 2016. ISSN 2050084X. doi: 10.7554/ELIFE.16965.

Jason Yang, Ravi G. Lal, James C. Bowden, Raul Astudillo, Mikhail A. Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H. Arnold. Active learning-assisted directed evolution. *Nature Communications 2025 16:1*, 16:1–12, 1 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-55987-8. URL https://www.nature.com/articles/s41467-025-55987-8.

Jason Z. Zhang, Xinting Li, Caixuan Liu, Hanlun Jiang, Kejia Wu, and David Baker. De novo design of ras isoform selective binders. *bioRxiv*, pp. 2024.08.29.610300, 9 2024. doi: 10.1101/2024.08.29.610300. URL https://www.biorxiv.org/content/10.1101/2024.08.29.610300v3https://www.biorxiv.org/content/10.1101/2024.08.29.610300v3.abstract.

A APPENDIX

A.1 SURROGATE MODELS, ACQUISITION FUNCTIONS, AND HYPERPARAMETER TUNING

We implemented several standard surrogate models used in previous benchmarks for protein engineering BO (Yang et al., 2025) and adapted them to our case (Table A.1).

Table A.1: Surrogate model architectures

Surrogate	Architecture	Activation
Random forest	Tree ensemble	N/A
Deep kernel GP	3-layer MLP + GP	ReLU
Gaussian process	RBF or Matérn kernel and Gaussian likelihood	N/A
Bayesian NN	3-layer BNN	ReLU
Dropout NN	3-layer MLP	ReLU
Ensemble NN	3-layer MLPs	ReLU

Similarly, we explored 4 commonly used acquisition functions (Table A.2). Subsequent work could further augment these, with protein-specific ones that could improve the optimization process. For example, we could use a classifier to predict the probability of binding along the binding affinity and use it in the acquisition, as done by Rapp et al. (2024), or augment it with evolutionary or structural priors (Frisby & Langmead, 2021). We believe more work could be done to define principled acquisitions for the binding improvement problem.

Table A.2: Acquisition functions implemented in our benchmark. $\mu(x)$ and $\sigma(x)$ are the predicted mean and standard deviation from the surrogate model, f^* is the current best-observed value, Φ and ϕ are the CDF and PDF of the standard normal distribution. We did not perform a hyperparameter search for the acquisition function.

Acquisition	Formula	Parameters
Expected improvement	$\mathbb{E}[\max(f(x) - f^* - \xi, 0)]$ $= (\mu(x) - f^* - \xi)\Phi(z) + \sigma(x)\phi(z)$ where $z = \frac{\mu(x) - f^* - \xi}{\sigma(x)}$	$\xi = 0.01$
Upper confidence bound	$\mu(x) + \beta\sigma(x)$	$\beta = 2.0$
Thompson sampling	$f(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$	None
Greedy	$\mu(x)$	None

For our optimization campaign, we conducted a comprehensive evaluation of model combinations to determine their peak performance and ensure our model comparisons as robust. We implemented a grid search across multiple hyperparameters, testing each parameter configuration on individual landscapes. In each landscape, we allocated 20% of the sequences for hyperparameter optimization (15% training, 5% testing), while reserving the remaining 80% for the actual simulated campaign. We chose a random split over a homology or a high-low activity one (Dallago et al., 2022) because we wanted to maintain the inherent fitness distribution and preserve the landscape topology. We selected our grid samples by consulting the literature, both for previous protein BO campaigns, the original surrogate implementations, and standardized implementations from libraries like GPyTorch (e.g., the number of Monte-Carlo samples or dropout rate as highlighted by Gal & Ghahramani (2015)).

We used a schedule-free implementation of the Adam optimizer for both the hyperparameter tuning and simulated campaigns, as initially described by Defazio et al. (2024).

Table A.3: Hyperparameter search space for each surrogate model. Common parameters across neural models: no. epochs = 100, batch size = 32, Monte Carlo samples = 30 (where applicable).

Surrogate	Hyperparameter search space
Random forest	no. estimators $\in \{10, 50, 100, 200\}$ max depth $\in \{None, 10\}$
Deep kernel GP	hidden dimension = 128 learning_rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}\}$ kernel_type $\in \{$RBF, Matérn$\}$
Gaussian process	kernel type $\in \{$RBF, Matérn$\}$ learning rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}\}$
Bayesian NN	hidden dimension = 128 learning rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}\}$ KL weight = 1.0
Dropout NN	hidden dimension = 128 learning rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}\}$ dropout rate = 0.1
Ensemble NN	hidden dimension = 128 learning rate $\in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}\}$ no. estimators = 5

A.2 Landscape property calculations

We calculated the properties of our landscapes following the methodology established by Li et al. (2024), with our main difference in using the Otsu method to determine the threshold for active variants. These are summarized in the table below. N represents the total number of variants. Ruggedness (Rugged.) quantifies local structure complexity. Peak fitness shows the Cauchy distribution peak location. Kurt. (kurtosis) and Skew (skewness) describe the fitness distribution shape. KDE peaks indicate the number of modes in the kernel density estimation. Optima shows the count of local fitness maxima. Magnitude (Mag.) and non-magnitude (Non-mag.) epistasis percentages quantify pairwise interaction types.

Table A.4: Landscape properties.

Dataset	Active% (Otsu)	Thresh. (Otsu)	N	Rugged.	Peak fitness	Kurt.	KDE peaks	Local optima	Mag. epist.	Non-mag. epist.	Skew.
KRAS	57.59	-0.56	19899	0.21	-0.38	-1.16	3	576	25.31	74.69	-0.36
GB1 subset	3.82	1.32	6080	1.73	0.00	36.53	14	7	6.71	93.29	5.61
SpikeRBD	82.40	-2.08	3042	0.90	-0.19	1.47	2	186	0.00	0.00	-1.64
CytP4502C9	51.01	0.53	4914	0.40	0.54	-1.41	2	465	0.00	0.00	0.07
CCR5	75.41	-0.37	4910	0.74	-0.02	5.18	4	323	0.00	0.00	-1.05
CD19	17.65	0.30	3009	4.24	-2.02	1.82	2	269	0.00	0.00	1.01
Gcn4	45.75	1.32	2111	0.30	1.29	16.89	6	47	0.00	100.00	1.02
ACE2	63.07	-0.66	1779	1.40	-0.25	0.72	2	117	0.00	0.00	0.19
HLA-A	65.73	-0.18	2676	1.35	0.22	0.91	2	178	0.00	0.00	-0.54
YAP1	21.45	1.29	8060	1.21	0.54	12.11	9	926	5.49	94.51	2.52
Dlg4 PDZ3	87.63	-0.59	1261	0.29	0.01	3.48	3	83	0.00	0.00	-2.03

A.3 OPTIMIZATION RISK, PERFORMANCE, AND COST CALCULATIONS

Cold-start quantification using the $\Delta G\ AUC$: We define the optimization strategy O_i as the combination of surrogate, acquisition, encoding, loss, and kernel for a given choice of these parameters i. We set the initial starting pool of variants as n^s_{init} for the initialization (seed) s and n^s_k as the current acquired variants at iteration k. In total, we set a fixed budget $n_{budget} = n_{init} + b * K$, where b is the batch size and K is the number of optimization cycles.

We define G^s_k as the payoff (reward, performance) achieved after conducting the optimization at iteration k accounting for O_i, n^s_{init}, and n^s_k. For single-objective optimization, G^s_k will be:

$$G^s_k(O_i, n^s_{\text{init}}, n^s_k) = \max_{x^s_k \in X} f(x^s_k) \mid O_i, n^s_{\text{init}}, n^s_k)$$

Where x^s_k contains all variants acquired including iteration k for seed s.

Thus, we are continuously aware of our budget and variants acquired at each iteration when computing the payoff. We have defined payoff as the maximum acquired binding affinity up until the kth iteration, but other metrics such as cumulative regret could be used.

We want to calculate the ΔG difference in optimum values for a given model choice O_i and a simple baseline O_0 for the early stages of optimization, for a given seed. The simple baseline will be a random search in our case, which mimics the approximate behaviour of a Bayesian optimization algorithm with uniformly distributed priors in the early stages of optimization to quantify cold starts.

$$\Delta G^s_k = G^s_k(O_i, n^s_{init}, n^s_k) - G^s_k(O_0, n^s_{init}, n^s_k)$$

Thus, cold starts are represented by a small ΔG^s_k: the model needs to learn more about the data to suggest optimal value variants, as it resembles a random search initially. A large ΔG^s_k suggests effective utilization of prior information, leading to better performance.

Furthermore, we can average ΔG^s_k across all seeds, with $\Delta G_k = \mathbb{E}[\Delta G^s_k \mid s \in S]$. Furthermore, we will compute the Area Under the Curve (AUC) to obtain $\Delta G^s_{1:K}\ AUC$ - the cumulative model's performance considering all time steps (k from 1 to K). This ensures we rank highly models that do not necessarily converge to the optimum, but achieve a steady improvement throughout all iterations (especially if we want to stop an optimization earlier if we are content with the results).

Risk metrics for ranking: We are interested in quantifying the performance in worst-case scenarios, and thus implemented risk measures from financial mathematics - VaR (value at risk) and CVaR (Conditional Value at Risk, also known as the Expected Shortfall, ES). (Artzner et al., 1999; Cakmak et al., 2020).

$$\text{VaR}_\alpha(X) = \inf\{x \in \mathbb{R} : P(X \leq x) \geq \alpha\}$$

$$\text{CVaR}_\alpha(X) = \mathbb{E}[X \mid X \leq \text{VaR}_\alpha(X)]$$

In our benchmark, we have used the CVaR/ES with an $\alpha = 0.1$ calculated for the final fitness performance, performance relative to baseline, and costs, determining the average of payoffs that do not reach the VaR threshold (fitness in the worst 10% of cases). To make mathematical notation less abstract, a large ES means that even in the worst 10% of initializations, a model will still yield a high payoff throughout all iterations (AUC), and will outperform a random baseline (positive AUC). An ES close to the maximum AUC means the model is robustly parsing the fitness landscape, irrespective of its initialization.

A.4 BOOTSTRAP ANALYSIS

Two bootstrap analyses were performed to address the limited number of runs (seeds) sampled.

A.4.1 COMPARING MEAN VERSUS CVAR-BASED MODEL RANKING

We performed a bootstrap analysis with increasing stringency to assess the statistical significance of the mean and CVaR model rankings and their associated campaign cost differences. For the naive bootstrap, we sampled with replacement from the available optimization runs (20 seeds) for each dataset-model pair. We then ranked the top models for each bootstrap sample using either the average fitness over the seeds or the CVaR, selected the best one for each strategy, and then compared the corresponding cost difference to reach the 99th percentile fitness between the CVaR and mean strategies. We did not split the runs into ranking and evaluation ones. This yielded some impressive (albeit overestimating benefits due to data leakage) results: CVaR-based ranking saved upwards of 25% of the optimization campaign, improved average outcomes, and had a negative trend only for a single complex landscape with highly variable campaign outcomes (Figure A.6 A).

Next, we implemented an out-of-bag bootstrap strategy: 80% of seeds were used for model ranking (16 seeds) and the remaining were reserved for evaluation. We then generated all possible combinations of ranking and evaluation seeds, evaluated the average and worst-case (of the evaluation seeds) cost savings, and calculated confidence intervals using the percentile method (Figure A.6 B and C). While naive bootstrapping suggested substantial benefits from risk-aware selection, more rigorous statistical testing revealed these differences were not significant enough to justify favouring one ranking method over the other. We tried to address the lower number of runs per model via the bootstrapping approach, yet this is still a limitation of our work.

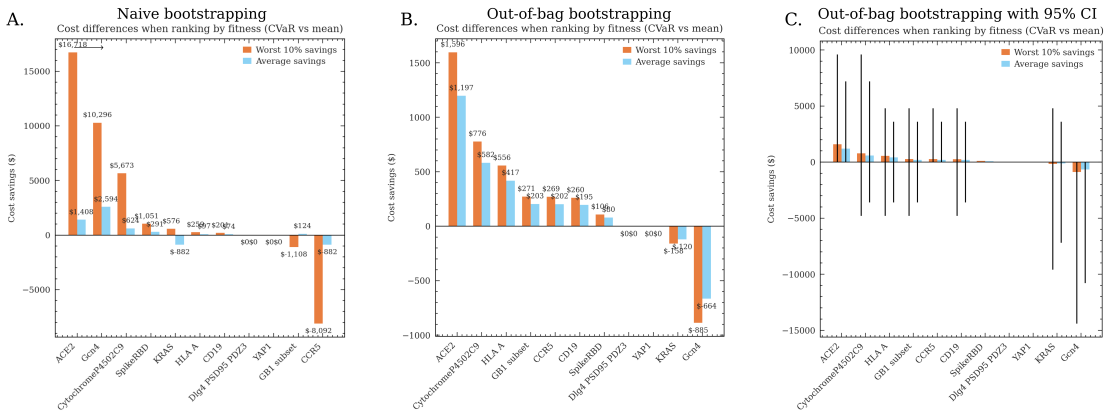

Figure A.6: Comparison of different bootstrapping and evaluation techniques to establish the cost savings between CVaR-based and mean-based model ranking.

A.4.2 ESTABLISHING THE CONFIDENCE OF CORRELATIONS BETWEEN LANDSCAPE PROPERTIES AND MODEL PERFORMANCES

To account for the limited number of runs per model type, we bootstrapped (sampling with replacement) the seeds, then calculated the CVaR and average model performance for our 3 metrics, averaged them for each landscape, and computed the Kendall τ correlation scores between the average model performance and landscape properties. This was done for 1000 bootstrap samples and confidence intervals were calculated with the percentile method. We observed similar confidence intervals for increasing bootstrap sample sizes. Our results are summarized below. Overall, the correlation coefficients between final fitness and properties are confidently established, yet they vary more when considering the cost metrics. We assume this is due to some sequences above the 99th threshold being acquired earlier simply by chance, even if the model cannot effectively exploit this fitness region.

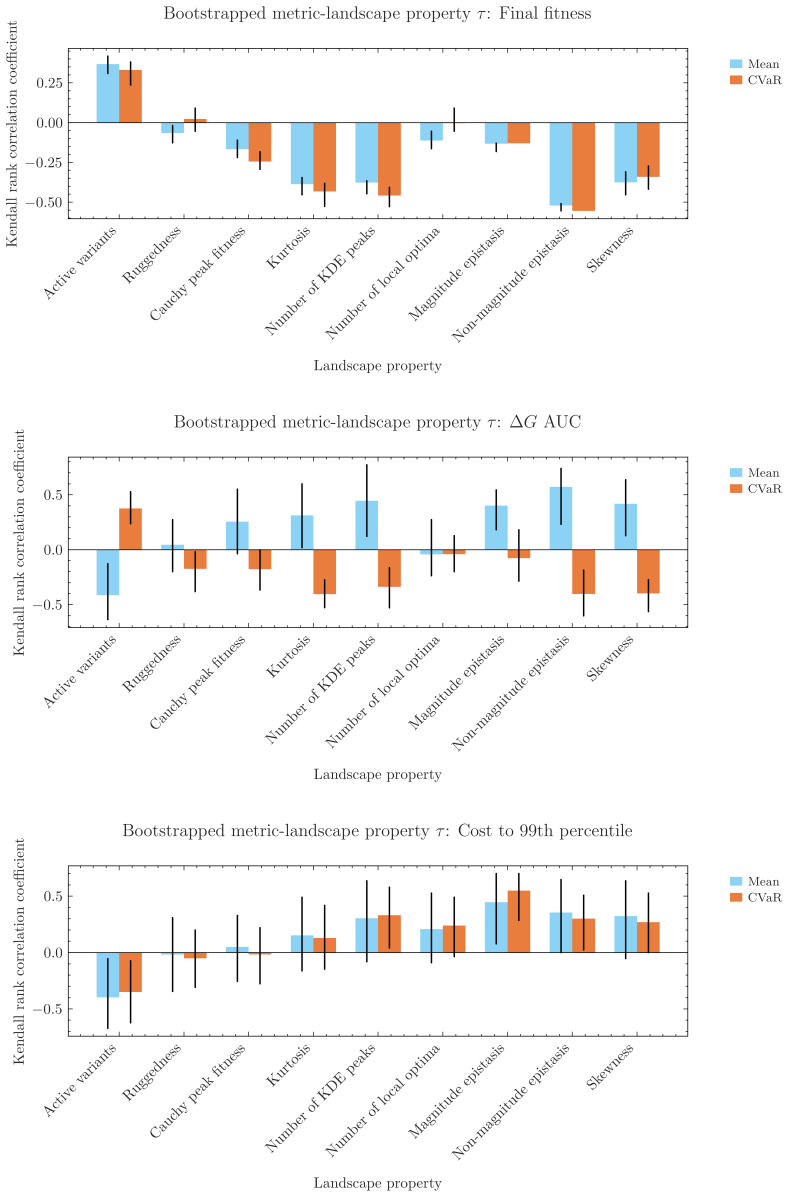

Figure A.7: Kendall τ values following the bootstrap analysis between the average model performance or risk and several landscape properties. Error bars indicate the 95% confidence intervals.

A.5 ADDITIONAL RANK CORRELATIONS

We have highlighted additional Kendall τ correlations between risk and average-based rankings for the $\Delta G\ AUC$ and cost metrics (Table A.5). Seemingly, models tend to require similar numbers of sequences regardless of ranking method to reach our fitness threshold, indicating it might either be too strict or easy to reach. This showcases the difficulty of setting *a priori* absolute optimization goals. Subsequent analyses could set an achievable fold-change goal instead. When considering the performance relative to a baseline some models that perform well on average struggle in worst cases (e.g., on the CCR5 landscape), suggesting the landscape may have regions where models fail to distinguish themselves from random search.

Table A.5: Rank agreement (as Kendall τ correlation) between CVaR and average-based model rankings. Statistical significance: $^{*}p < 0.05$, $^{**}p < 0.01$ $^{***}p < 0.001$.

Dataset	ΔG AUC rank correlation	Cost to 99th percentile rank correlation
HLA-A	0.519^{***}	0.937^{***}
CD19	0.487^{***}	0.919^{***}
CCR5	0.295^{***}	0.933^{***}
ACE2	0.677^{***}	1.000^{***}
CytochromeP4502C9	0.591^{***}	0.759^{***}
Dlg4 PSD95 PDZ3	0.588^{***}	0.971^{***}
KRAS	0.546^{***}	0.703^{***}
GB1 subset	0.720^{***}	0.834^{***}
YAP1	0.677^{***}	0.581^{***}
SpikeRBD	0.475^{***}	0.976^{***}
Gcn4	0.369^{***}	0.913^{***}
All datasets	0.529^{***}	0.816^{***}

A.6 RANK AGREEMENT VERSUS LANDSCAPE PROPERTY CORRELATIONS

We further analysed which landscape properties influence the rank correlation between CVaR and average-based methods. We did not observe any statistically significant results.

Table A.6: Correlation between landscape properties and rank agreement between average and worst-case (CVaR) metrics. Statistical significance: $^{*}p < 0.05$, $^{**}p < 0.01$ $^{***}p < 0.001$.

Landscape property	Final fitness rank correlation	ΔG AUC rank correlation	Cost to 99th percentile correlation
Active variants % (Otsu)	-0.422	0.000	-0.467
Ruggedness	-0.156	-0.135	0.156
Cauchy peak fitness	-0.180	-0.432	0.225
Kurtosis	0.022	-0.360	-0.333
Number of KDE peaks	0.227	0.102	0.025
Number of local optima	-0.244	0.000	0.244
Magnitude epistasis	0.243	-0.062	0.243
Non-magnitude epistasis	0.109	-0.028	0.218
Skewness	0.067	0.360	-0.289

A.7 Optimization curves for the top models across different landscapes

A.7.1 GB1 top models

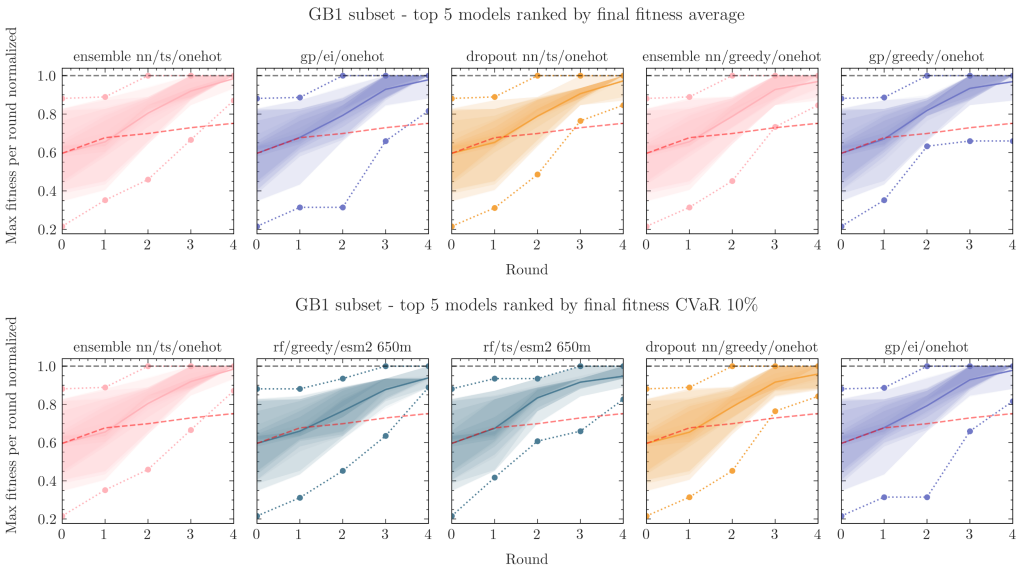

Figure A.8: Top 5 models for the average and CVaR-based selection on the final fitness reached - GB1 subset dataset.

A.7.2 CCR5 top models

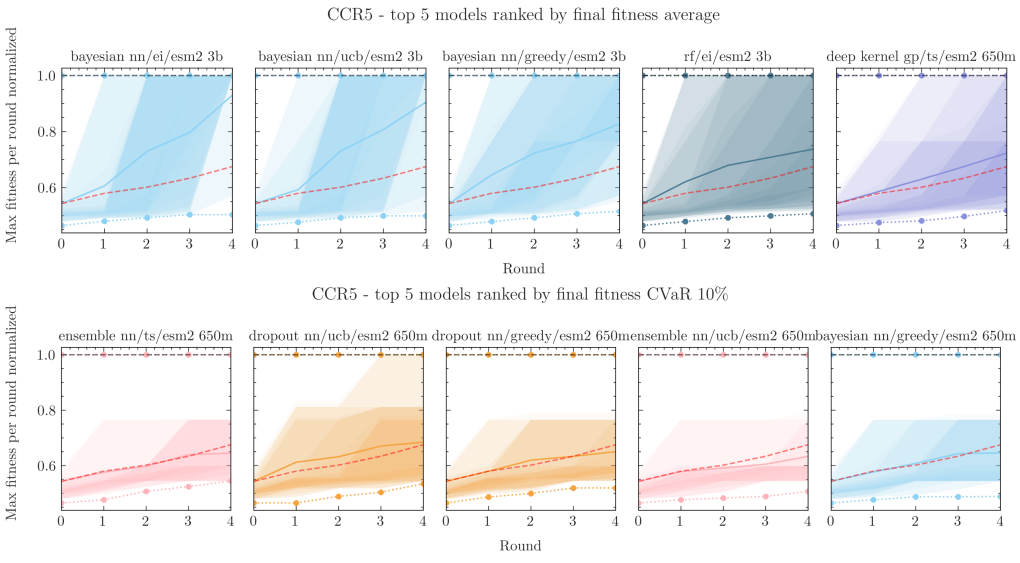

Figure A.9: Top 5 models for the average and CVaR-based selection on the final fitness reached - CCR5 dataset.

A.7.3 CD19 TOP MODELS

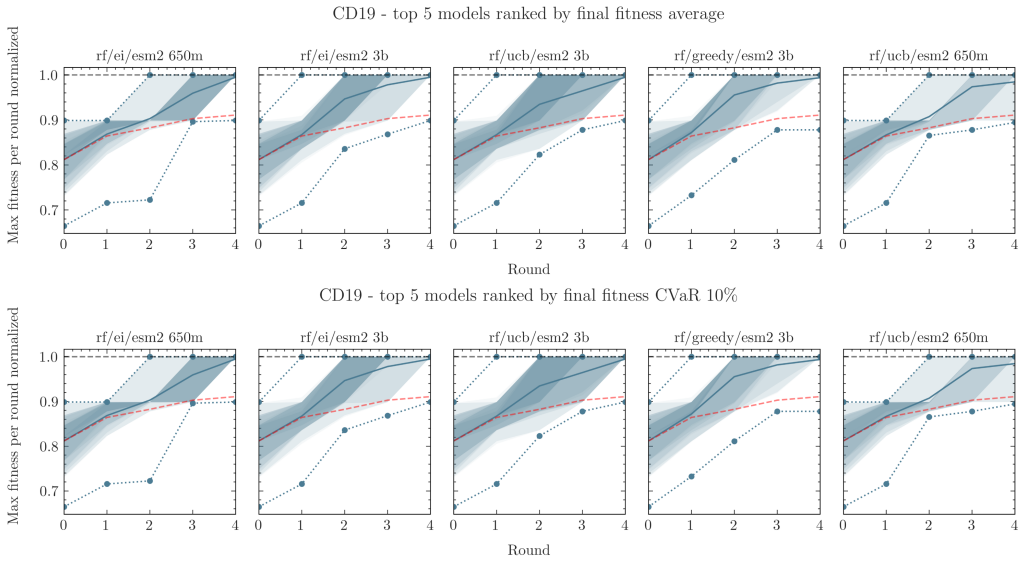

Figure A.10: Top 5 models for the average and CVaR-based selection on the final fitness reached - CD19 dataset.

A.7.4 HLA-A TOP MODELS

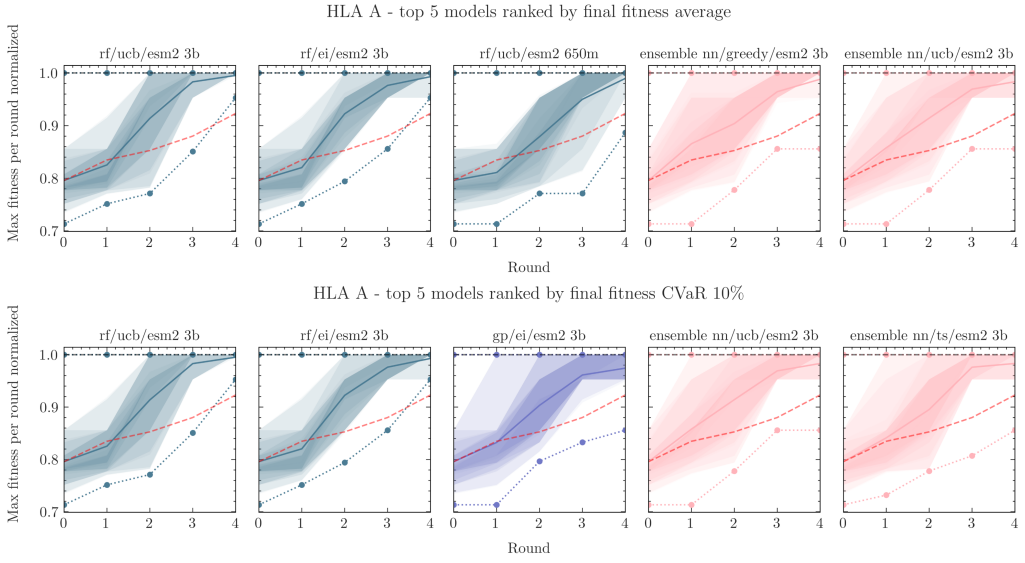

Figure A.11: Top 5 models for the average and CVaR-based selection on the final fitness reached - HLA-A dataset.

A.7.5 CYTOCHROME P4502C9 TOP MODELS

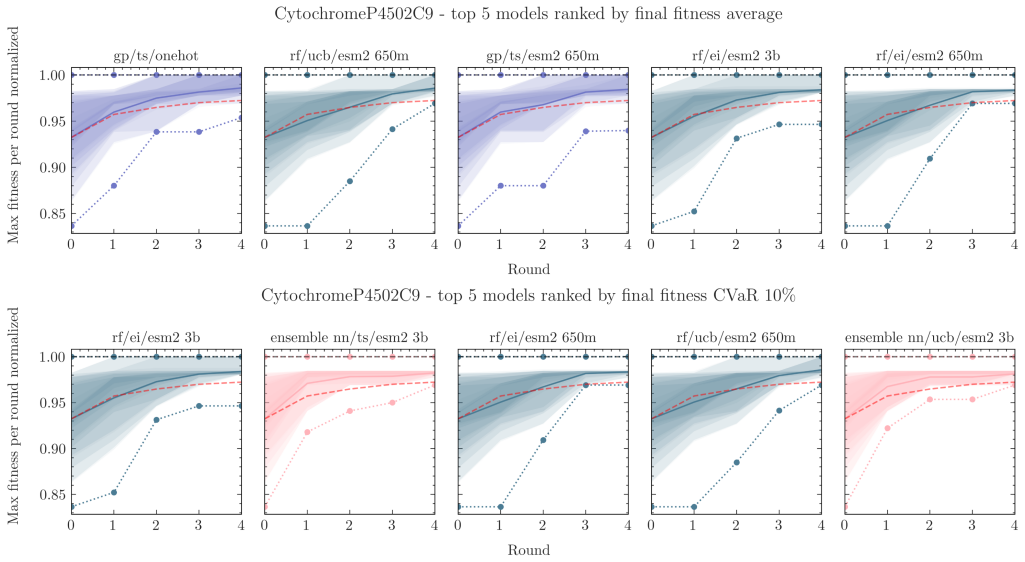

Figure A.12: Top 5 models for the average and CVaR-based selection on the final fitness reached - Cytochrome P4502C9 dataset.

A.7.6 KRAS1 TOP MODELS

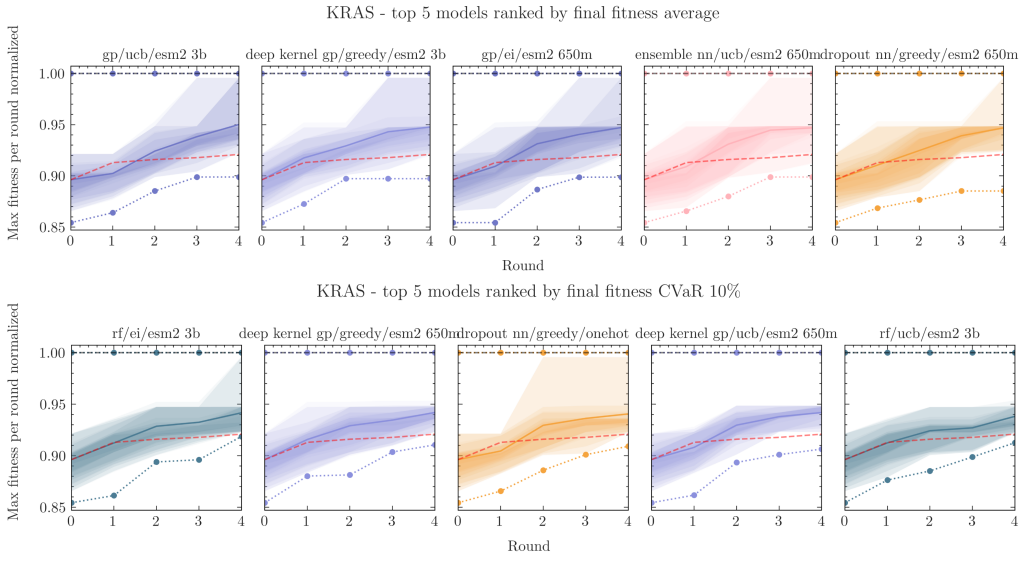

Figure A.13: Top 5 models for the average and CVaR-based selection on the final fitness reached - KRAS1 dataset.

A.7.7 SPIKERBD TOP MODELS

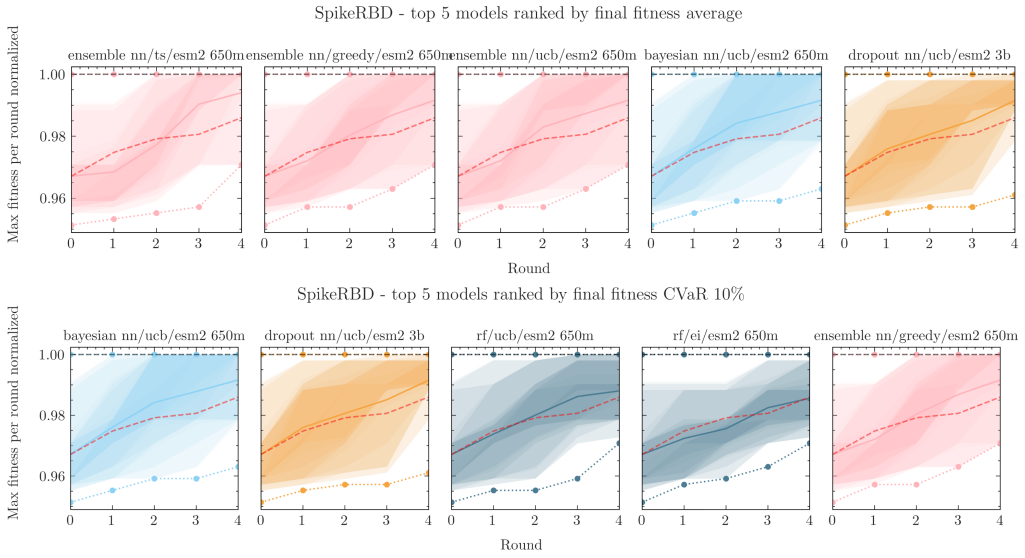

Figure A.14: Top 5 models for the average and CVaR-based selection on the final fitness reached - SpikeRBD dataset.

A.7.8 DLG4 TOP MODELS

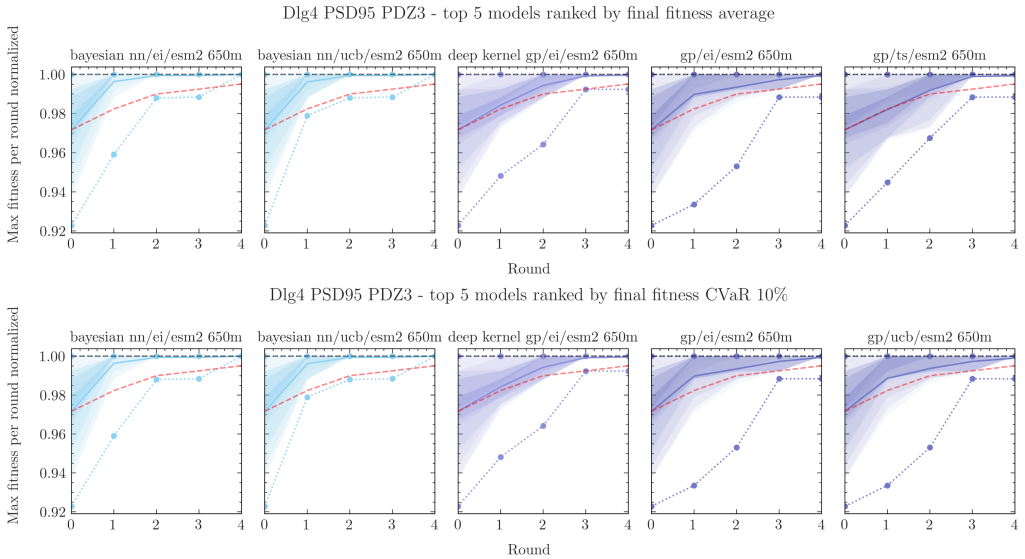

Figure A.15: Top 5 models for the average and CVaR-based selection on the final fitness reached - Dlg4 dataset.

A.7.9 GCN4 TOP MODELS

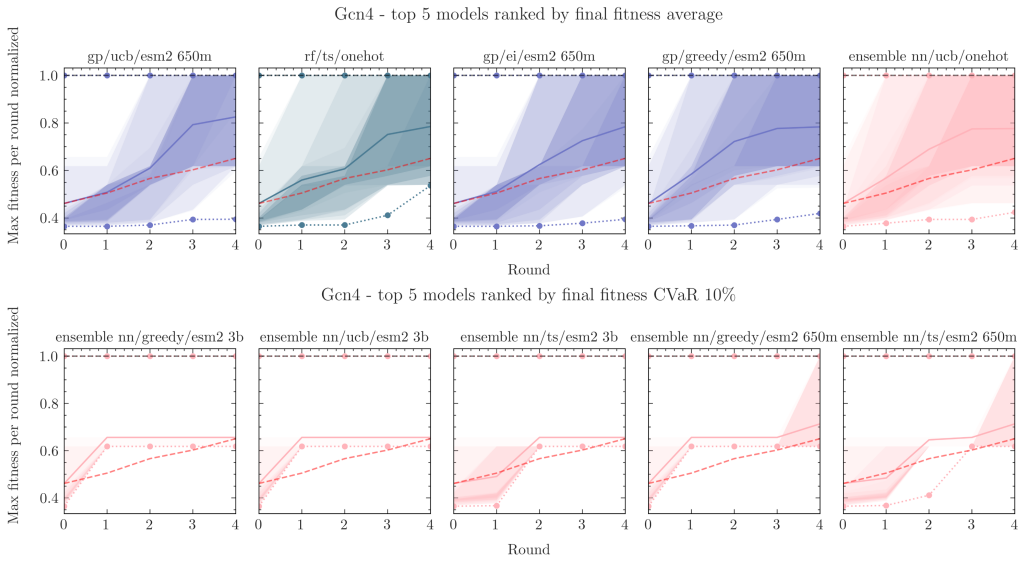

Figure A.16: Top 5 models for the average and CVaR-based selection on the final fitness reached - Ccr5 dataset.

A.7.10 ACE2 TOP MODELS

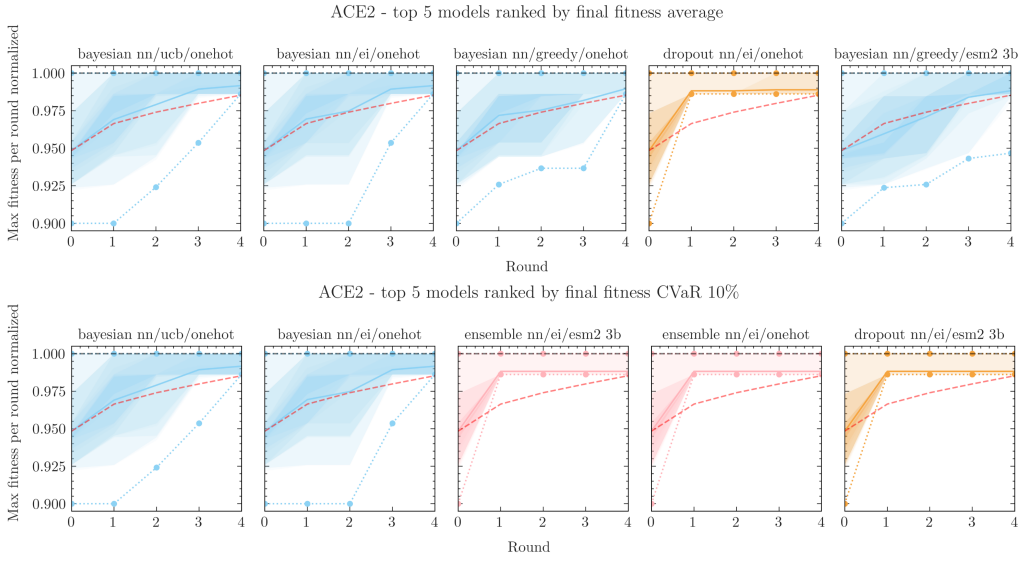

Figure A.17: Top 5 models for the average and CVaR-based selection on the final fitness reached - ACE2 dataset.

A.7.11 YAP1 TOP MODELS

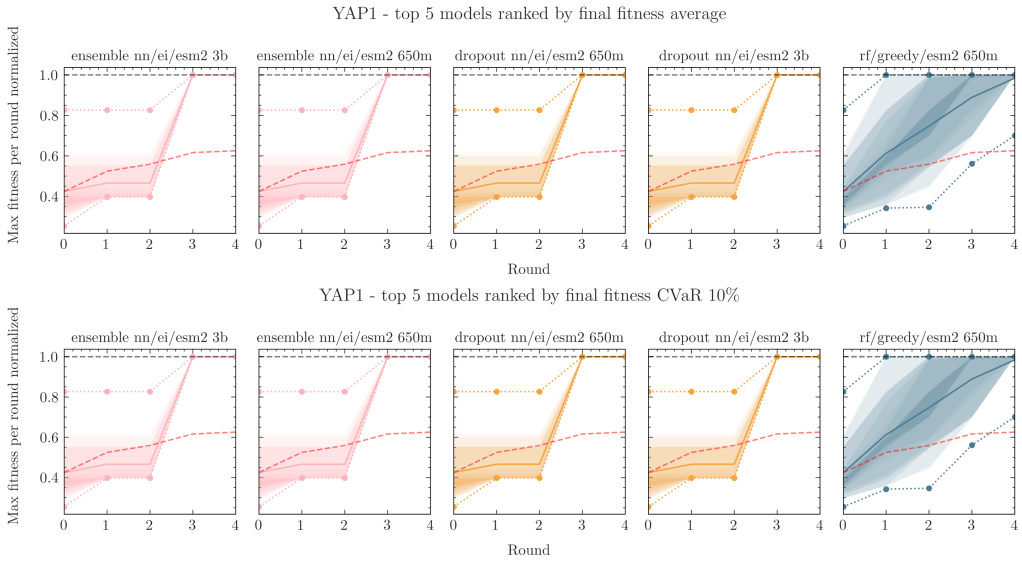

Figure A.18: Top 5 models for the average and CVaR-based selection on the final fitness reached - YAP1 dataset.

A.8 COST-PERFORMANCE PARETO FRONTS FOR THE GB1 SUBSET DATASET

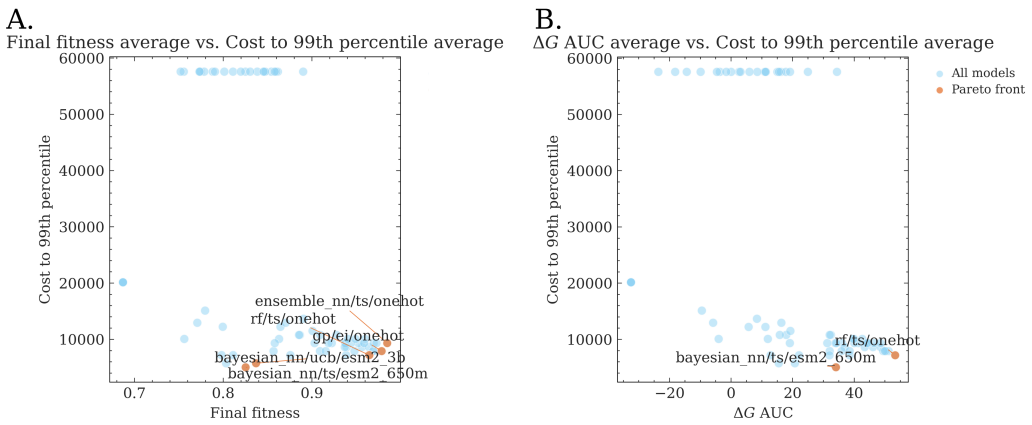

Figure A.19: Performance (as the average final fitness reached and average $\Delta G\ AUC$) versus costs to reach the 99th fitness percentile. We have highlighted the Pareto-optimal models.